

**ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA
SEDE DI CESENA**

**FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E
NATURALI**

CORSO DI LAUREA IN SCIENZE DELL'INFORMAZIONE

Titolo della tesi

**PROGETTAZIONE E REALIZZAZIONE DI UN
PORTALE WEB PER ANALISI STATISTICA
RELATIVA ALL'ATTIVITÀ SVOLTA IN
RADIOTERAPIA**

Tesi di laurea in

SISTEMI MULTIMEDIALI

Relatore

Prof.ssa Paola Salomoni

Presentata da

Andrea Zaccheroni

Sessione III°
Anno Accademico 2011-2012

INDICE

INTRODUZIONE	1
CAPITOLO 1. PREMESSE	5
1.1 Attività di Radioterapia.....	5
1.2 Biostatistica in Radioterapia	6
1.2.1 Analisi di sopravvivenza.....	7
1.2.2 Funzioni equivalenti per l'analisi della sopravvivenza.....	9
1.2.3 I metodi	10
1.2.4 Il metodo del prodotto limite (Kaplan-Meier)	11
1.2.5 Il confronto di due curve di sopravvivenza (logrank test)	13
1.3 Sistemi Informatici Radioterapia IRST	15
1.3.1 Cartella clinica ospedaliera	16
1.3.2 Cartella clinica di reparto.....	17
1.4 Estrazione e presentazione dei dati clinici e amministrativi.....	18
CAPITOLO 2. PROGETTAZIONE E TECNOLOGIE UTILIZZATE.....	21
2.1. Analisi dei software e dei database presenti	21
2.1.1 Software Amministrativo Ospedaliero	21
2.1.2 Analisi database amministrativo ospedaliero	25
2.1.3 Software clinico e amministrativo radioterapia	27
2.1.4 Analisi database Mosaiq	32
2.2 Progettazione delle applicazioni e dei database.....	37
2.2.1 Gestioni permessi utenti.....	38
2.3 Progettazione Interfaccia Utente.....	41
2.3.1 Progettazione autenticazione utente.....	41
2.3.2 Progettazione portale radioterapia	44
2.3.3 Progettazione statistiche amministrative	44
2.3.4 Progettazione statistiche di reparto (ambulatori e macchinari)	45

2.3.5 Progettazione Curve di Sopravvivenza	47
2.4 Tecnologie utilizzate per lo sviluppo	50
2.4.1 PHP.....	50
2.4.2 AJAX.....	54
2.4.3 JSON	55
2.4.4 XML e XHTML	58
2.4.3 JavaScript e JQuery	60
CAPITOLO 3. IMPLEMENTAZIONE	63
3.1 Modulo gestione Utenti	63
3.1.1 Set funzioni PHP	63
3.1.2 Logica di autenticazione.....	67
3.1.3 Pagina di login e recupero password.....	68
e riceve un oggetto JSON così strutturato:	72
3.1.4 Barra applicazione	72
3.2 Modulo statistiche Amministrative	73
3.3 Modulo statistiche di Reparto	76
3.4 Modulo Curve di Sopravvivenza.....	80
3.4.1 Interfaccia di selezione dei casi da analizzare.....	80
3.4.2 Interfaccia di post elaborazione.....	82
3.4.3 Visualizzazione grafica interattiva dei risultati	84
3.4.4 Strutture dati XML e JSON.....	85
CAPITOLO 4. CONCLUSIONI E SVILUPPI FUTURI.....	89
BIBLIOGRAFIA.....	91

INDICE DELLE FIGURE

Figura 1.1 - Esempio di Curva di Kaplan Meyer.....	13
Figura 2.1 - Schermata agende LOG80	22
Figura 2.2 - Schermata accessi paziente LOG80.....	23
Figura 2.3 - Schermata registrazione e controllo prestazioni effettuate	24
Figura 2.4 - Relazioni tabelle per vista LOG80.....	27
Figura 2.5 - Schermata Location Mosaiq.....	28
Figura 2.6 - Esito della seduta di terapia Mosaiq	29
Figura 2.7 - Finestra Diagnoses and Interventions Mosaiq	30
Figura 2.8 - Diagnosis and Staging Mosaiq.....	31
Figura 2.9 - Finestre Follow-Up summary e Follow-Up Mosaiq.....	29
Figura 2.10 - Schema del database IRST per l'autenticazione degli utenti.....	40
Figura 2.11 - Mockup schermata di login utente	42
Figura 2.12 - Barra utente presente in ogni applicazione	43
Figura 2.13 - Menu per la funzionalità di cambio profilo	44
Figura 2.14 - Statistiche amministrative - Sezione Monitor.....	45
Figura 2.15 - Statistiche Ambulatori e Diagnostiche.....	46
Figura 2.16 - Filtro per Curve di sopravvivenza.....	47
Figura 2.17 - Elaborazione estrazione Curve di sopravvivenza	48
Figura 2.18 - Visualizzazione dati estratti Curve di sopravvivenza	49
Figura 2.19 - Oggetto JSON	56
Figura 2.20 - Array JSON.....	57
Figura 2.21 - Tipi di valore in JSON	57
Figura 2.22 - Tipo numero JSON	57
Figura 3.1 - Figura 3.1 - Login IRST.....	68
Figura 3.2 - Utente o password non riconosciuti	69
Figura 3.3 - Modifica Password.....	70

Figura 3.4 - Recupera Password dimenticata	70
Figura 3.5 - Barra Applicazione Utente non abilitato al programma.....	72
Figura 3.6 - Tendina opzioni utente	73
Figura 3.7 - Analisi dati aggregati prestazioni radioterapia	74
Figura 3.8 - Analisi dati aggregati prestazioni radioterapia 2	75
Figura 3.9 - Monito ambulatori e diagnostiche	76
Figura 3.10 - Schermata di attesa	77
Figura 3.11 - Dettaglio appuntamenti diagnostica	79
Figura 3.12 - Datepicker e intervallo di tempo maggiore	79
Figura 3.13 - Menu curve di sopravvivenza.....	80
Figura 3.14 - Esempio di estrazione.....	81
Figura 3.15 - Menu post-elaborazione	83
Figura 3.16 - Dati curve	83
Figura 3.17 - Grafico con raggruppamento	84

INTRODUZIONE

L'avvento dell'Information Technology all'interno di realtà medico-sanitarie ha determinato notevoli cambiamenti nel modo in cui i dati vengono gestiti, l'evoluzione delle tecnologie per la gestione delle informazioni ha segnato profondamente non solo la parte amministrativa ma ha anche contribuito all'evoluzione e al miglioramento delle pratiche mediche. Le note sulle condizioni dei pazienti e le procedure assistenziali applicate, che costituiscono il patrimonio informativo delle strutture che operano in questo campo, sono da sempre state archiviate su carta. L'informatizzazione consente, allo stato attuale, di disporre di sistemi automatici per la memorizzazione e la ricerca di dati.

Molto spesso però questi sistemi, probabilmente a causa di un'origine non prettamente ospedaliera, non offrono una soluzione completa a quelli che possono essere i problemi legati alla gestione di un'attività di reparto complessa e costituita da elementi eterogenei. Ancora di più in un centro di ricerca medica dove si tenta costantemente di migliorare i propri standard, tenere sotto controllo efficienza e risultati di diverse procedure simultaneamente risulta essere particolarmente complesso ed emerge nei sistemi attuali una mancanza di flessibilità che richiede necessariamente l'utilizzo di applicazioni terze. Dalle problematiche più semplici come la gestione dell'occupazione di una risorsa ospedaliera fino ai contesti più rilevanti, che hanno l'obiettivo di migliorare la qualità del servizio offerto, nasce l'esigenza di monitorare aspetti che toccano campi diversi.

La sperimentazione può essere definita come uno studio condotto su persone sofferenti di una malattia per ottenere una migliore comprensione delle cause, dell'evoluzione e delle modalità di trattamento della malattia stessa e con l'obiettivo finale di migliorare la longevità e la qualità della vita dei malati presenti e futuri. I metodi di ricerca utilizzati nella ricerca clinica possono essere distinti in due grandi gruppi: studi sperimentali che comprendono gli studi clinici controllati utilizzati nella valutazione dell'efficacia di farmaci ed altri interventi medici; studi osservazionali che possono essere descrittivi o analitici. Parte cruciale nella sperimentazione clinica è ricoperta dalla statistica medica che si pone il fine di verificare l'efficacia o meno di terapie o

scoprire se un qualche problema medico possa dipendere da fattori particolari. Purtroppo non è facile trovare un set di dati minimi che permettano l'analisi di tutti gli studi clinici e anche l'informatizzazione della cartella clinica non risponde mai al cento per cento alle esigenze statistiche.

Nelle strutture ospedaliere operano quindi reparti di biostatistica che gestiscono la raccolta e l'analisi dei dati, coadiuvati da soluzioni software per la statistica conformi all'elaborazione dei dati clinici. L'interazione tra medici e statistici porta a una consapevolezza maggiore dell'effetto del lavoro svolto sul paziente con un conseguente miglioramento delle pratiche applicate. Questo implica però un maggiore impiego di risorse, anche umane che potrebbero in alcuni casi rallentare il processo di ricerca stesso. Non bisogna comunque dimenticare che in queste strutture viene portato avanti anche un servizio assistenziale che prescinde dalla ricerca e che è sottoposto dagli enti certificatori a controlli di qualità molto selettivi, nasce quindi l'esigenza per la direzione di poter consultare indicatori di qualità del servizio offerto.

In questo contesto lo scopo che la tesi si pone è quello di realizzare un'applicazione in grado di avvicinare e facilitare la consultazione di indicatori per il personale medico e di direzione, relativi in questo caso specifico all'unità operativa di radioterapia presso l'istituto romagnolo per lo studio e la cura dei tumori, IRST di Meldola. I valori rappresentati devono permettere di valutare in tempo reale l'andamento del servizio e tener traccia dell'attività svolta sotto diversi punti di vista al fine di migliorare le politiche di gestione e le procedure mediche in uso.

Per realizzare questo obiettivo, sono state raccolte e poi sviluppate, le idee di chi deve, in prima persona, svolgere questo tipo di compito: la direzione sanitaria è stata coinvolta per quanto riguarda la parte prettamente amministrativa e cioè collegata principalmente ai costi di gestione; alla direzione medica di reparto è stata invece chiesto di considerare l'utilizzo dei dati potenzialmente presenti nei software utilizzati in reparto per la realizzazione di qualcosa di innovativo che, partendo dall'analisi del lavoro svolto potesse fornire informazioni utili per valutarlo fino a quel momento in maniera semplice e immediata. Come accennato non è solo il medico ad occuparsi della valutazione del reparto, nel merito entra anche l'unità di biostatistica che fornisce le capacità e la conoscenza necessaria a tale fine.

La presenza di un informatico nella gestione di un ospedale o come in questo caso nella gestione del reparto di radioterapia, è una figura relativamente nuova, che offre competenze estranee, o in alcuni casi sottovalutate, non solo al personale di formazione medica ma anche a quello di formazione principalmente matematica come il biostatistico. Nel caso specifico si è pensato di apportare una visione più dinamica e di facile utilizzo per la consultazione delle informazioni in maniera rapida e sempre attuale, si è quindi stabilito di utilizzare gli elementi caratterizzanti del web 2.0 per raggiungere lo scopo. Infine, si è deciso di utilizzare le tecnologie più diffuse e moderne nel settore della programmazione web, cercando di ottenere, il più possibile, continuità e soprattutto compatibilità, con quelli che sono i sistemi utilizzati all'interno dell'ospedale.

Il seguito della tesi è organizzato in 3 capitoli che si articolano come segue.

- Il capitolo 1 ci introduce all'interno del reparto di radioterapia e esamina in maniera non troppo dettagliata quelle che sono le principali attività svolte in reparto. Da questa breve analisi si arriva ad esaminare i sistemi informatici adottati in reparto fino alle tecniche utilizzate in statistica per l'osservazione e la valutazione dei risultati ottenuti in uno studio clinico.
- Nel capitolo 2 si scende più in dettaglio nell'analisi dei sistemi informatici allo scopo di progettare le sorgenti di dati per l'applicazione, e conseguentemente si prosegue con la progettazione delle funzionalità e dell'aspetto da ottenere tramite l'interfaccia grafica. Definite le specifiche dell'applicativo, si procede all'analisi delle tecnologie da impiegare.
- Nel capitolo 3 è presentata l'applicazione, modulo per modulo, considerando i set di funzioni lato *server* e lato *client* che permettono la corretta gestione delle informazioni e le soluzioni adottate nella realizzazione dell'interfaccia grafica.

CAPITOLO 1.

PREMESSE

1.1 Attività di Radioterapia

La radioterapia è una disciplina clinica che si serve delle radiazioni per la cura di diverse malattie, fra le quali i tumori.

Le scoperte che stanno alla base della tecnica sfruttata dalla radioterapia risalgono alla fine del XIX secolo, e sono state rese possibili grazie agli studi di scienziati illustri come: Röntgen, Becquerel e i coniugi Curie.

Questi scienziati, insigniti del premio Nobel per le loro rispettive ricerche, scoprirono i raggi X e la radioattività naturale sulle cui basi si sono sviluppate le moderne tecniche di radioterapia [FLE81].

Tra queste le radiazioni ionizzanti sono una categoria fisica in grado di danneggiare le cellule neoplastiche; tali cellule tumorali che prima vengono colpite e danneggiate da queste radiazioni, successivamente sono distrutte ed eliminate dall'organismo stesso.

Circa la metà dei pazienti affetti da tumore deve sottoporsi a una qualsiasi forma di radioterapia nel corso della terapia antitumorale: per alcuni di loro la radioterapia è effettuata come trattamento curativo esclusivo per gli altri in combinazione con l'intervento chirurgico e/o la chemioterapia, si spera che il trattamento curi la neoplasia, o eliminandola o prevenendone le recidive, la radioterapia può essere sfruttata anche come trattamento palliativo, che non cura cioè il tumore ma si limita ad alleviare i sintomi e il dolore [FLE81].

Le radiazioni possono essere somministrate da un' apparecchiatura esterna che circonda il corpo del paziente (radioterapia esterna) oppure possono essere prodotte da una sostanza radioattiva che entra nell'organismo del paziente (brachiterapia).

Il tipo di radioterapia prescritta dall'oncologo dipende da diversi fattori, tra cui: il tipo di tumore, le dimensioni e la posizione dello stesso che incidono notevolmente nella pianificazione della terapia per via della vicinanza di tessuti sani sensibili alle radiazioni o meno.

La radioterapia esterna viene somministrata nella maggior parte dei casi sotto forma di fasci di fotoni, raggi X o Gamma.

I diversi tipi di Radioterapia esterna vengono effettuati usando un'apparecchiatura detta acceleratore lineare o LINAC la quale usa elettricità per formare fasci di particelle subatomiche accelerate, originando una radiazione molto potente [ELE10a].

Il paziente di solito riceve la radioterapia esterna in diverse sessioni giornaliere per un periodo di tempo che dipende dalla dose totale di radiazioni da somministrare.

Per mettere a punto il piano terapeutico del paziente il radioterapista si avvale di tecnologie particolarmente avanzate: TAC, MRI e PET servono per avere immagini digitali accurate dell'organismo del paziente, partendo da queste immagini, mediante l'utilizzo di software sofisticati che permettono la simulazione e il calcolo delle radiazioni assorbite dall'organismo [IRS12a].

Con l'aiuto di tecnici e fisici, il medico procede al contornamento dei "target", ovvero le zone da trattare e degli organi circostanti sensibili alle radiazioni, decide la quantità di radiazioni da somministrare (Gray Gy) e come indirizzare i fasci in modo opportuno.

Nonostante l'accuratezza dei sistemi computerizzati per la simulazione, durante la terapia vengono effettuati diversi controlli per accertarsi che la dose di radiazioni somministrata sia corrispondente a quella calcolata [IRS12a].

Per minimizzare i danni ai tessuti sani le radiazioni vengono frazionate in monodosi giornaliere, ed è per questo motivo che la terapia si protrae per diversi giorni.

A seguito della terapia il paziente deve essere sottoposto a controlli periodici, generalmente semestrali, che hanno lo scopo di monitorare lo stato della malattia ed evidenziare eventuali progressioni o recidive [IRS12a].

1.2 Biostatistica in Radioterapia

Il ricorso ai metodi statistici nel campo della ricerca medico-clinica è di rilevanza notevole, fin dai primi anni '60 infatti, con l'istituzione di nuove regole per le case farmaceutiche, veniva imposto per ogni nuovo prodotto presentato sul mercato l'accompagnamento di una notevole mole di dati raccolti a testimonianza della sicurezza ed efficacia dello stesso.

Per questo motivo ebbe inizio il ricorso sistematico e strategico dell'uso della statistica in medicina.

Dato il carattere multidisciplinare della ricerca clinica, il biostatistico dovrebbe prendere parte a tutte le fasi di un progetto e deve essere in grado di interloquire con i clinici al

momento dell' impostazione del problema e nella fase di interpretazione dei risultati, è necessario che abbia competenze sulle principali tecniche statistiche e i più importanti principi di epidemiologia e studi clinici.

La figura del biostatistico così inserita nel gruppo di lavoro, oltre a spiegare i risultati dello studio avrà anche un ruolo didattico per gli altri membri, elevando il loro livello di competenza statistica e producendo un effetto sinergico per il miglioramento nella ricerca.

All'interno di un gruppo di ricercatori coinvolti in uno studio clinico, almeno un componente deve avere competenze di informatica, molto spesso di questo ruolo si fa carico lo statistico altre volte è richiesta la competenza di un'ingegnere informatico [IRS12b].

Vengono utilizzate principalmente tre categorie di software: programmi di database per l'immissione dei dati, l'aggiornamento e semplici elaborazioni descrittive; pacchetti di programmi statistici per le più sofisticate, ma essenziali, analisi di routine tra cui le analisi di sopravvivenza; e linguaggi di programmazione per sopperire alle sempre meno presenti lacune dei programmi statistici.

Nel caso della radioterapia il reparto di biostatistica è stato coinvolto assieme al servizio informatico per effettuare un'analisi dei dati da raccogliere e individuare come sfruttare il software gestionale clinico presente in reparto con lo scopo di facilitare l'inserimento dei dati da parte del personale medico, l'unico con le competenze necessarie per questo tipo di lavoro, e allo stesso tempo rendere i dati reperibili in tempo reale al fine di realizzare analisi statistiche della sopravvivenza [NAN98].

1.2.1 Analisi di sopravvivenza

Nelle sperimentazioni cliniche condotte sulle malattie croniche potenzialmente fatali (ad es. tumori), la valutazione del risultato della terapia consiste nel fatto che il paziente sia morto o no (recidivato o no) e sul tempo trascorso dall'inclusione nello studio alla morte (o recidiva) o alla fine dell'osservazione.

Tale valutazione richiede l'utilizzo di tecniche di analisi della sopravvivenza.

I dati di sopravvivenza sono costituiti da una variabile di stato ed una variabile di risposta che misura il tempo che intercorre tra l'inizio dell'osservazione e la comparsa di un determinato evento, ritenuto rilevante e altre variabili che si suppone influenzino la variabile di risposta.

Lo scopo primario dell'analisi della sopravvivenza è quindi di descrivere la distribuzione della variabile di risposta e la sua relazione con altre variabili: continue come l'età o classificatorie come il sesso.

In altre parole, l'analisi della sopravvivenza effettua una stima della probabilità di avere un evento (morte, recidiva, progressione) dopo un determinato periodo di osservazione e pertanto la variabile oggetto di studio sarà il tempo che intercorre tra l'inizio dell'osservazione ed il verificarsi dell'evento [POC83].

Da un punto di vista formale, si tratta della realizzazione di una variabile casuale T che assume valori non negativi e la cui distribuzione è asimmetrica positiva.

Caratteristica peculiare dei dati di sopravvivenza è la presenza di *censura* in quanto il tempo esatto di comparsa dell'evento può non essere noto per tutti i pazienti; l'osservazione è quindi incompleta. Il caso più comune di osservazione incompleta è dovuto alla *censura a destra*. Una situazione tipica è rappresentata da uno studio che è interrotto quando è trascorso un tempo prefissato dalla data di inizio dell'osservazione; in questo caso non tutti i pazienti presentano l'evento di interesse entro il tempo fissato, e tutti i soggetti che non presentano l'evento stesso hanno tempi di osservazione uguali alla lunghezza dello studio (*censura di tipo I*). Altre situazioni di censura a destra si possono verificare nel caso di studi interrotti dopo che i pazienti hanno sviluppato l'evento di interesse (*censura di tipo II*) oppure nel caso in cui gli individui possono incorrere in eventi diversi dall'evento di interesse che causano la loro rimozione dallo studio (ad esempio non accessibilità ai controlli ambulatoriali, morte accidentale, ecc.) (*censura casuale*). Si assume quindi che per ogni individuo ci siano due variabili casuali: il tempo all'evento (Y_i) e il tempo di censura (T_{ci}) e quello che in effetti si osserva è una realizzazione delle variabili casuali:

$$T_i = \min (Y_i, T_{ci}) \quad i=1, \dots, n$$

con T_i tempo di osservazione [ARM71].

L'informazione fornita da ogni soggetto è sintetizzata nei due valori assunti rispettivamente dalla variabile casuale T_i e da un indice binario δ_i riguardante lo stato del paziente. δ_i è uguale ad 1 se l'evento è accaduto e 0 altrimenti.

La presenza di censura rende di fatto non utilizzabili i metodi di regressione usuali (ad es. modello lineare generale con distribuzione gaussiana) e si deve ricorrere a metodi sviluppati ad hoc. Un assunto alla base di tali metodi di analisi è che le variabili casuali

Y_i e T_{ci} siano indipendenti, ovvero che i pazienti con tempi censurati (o troncati) abbiano la stessa esperienza di vita dei pazienti per i quali è occorso l'evento.

Quando si pianifica una sperimentazione clinica si dovrà quindi decidere quale tipo di analisi della sopravvivenza effettuare, in quanto le tipologie possibili sono diverse; la differenza tra ciascun tipo è determinata da tre fattori: la popolazione su cui effettuare l'analisi, la definizione del tipo di evento da considerare e la definizione dell'intervallo di tempo relativo.

1.2.2 Funzioni equivalenti per l'analisi della sopravvivenza

Definiamo alcune funzioni dei dati di sopravvivenza e i relativi legami.

Sia S la **funzione di sopravvivenza** che misura la probabilità di un individuo di sopravvivere più a lungo del tempo t ed è così definita:

$$S(t) = p(T > t)$$

$$S(t) = 1 - F(t)$$

dove $F(t)$ è la distribuzione della variabile casuale T e

$$d F(t) / dt = f(t)$$

dove $f(t)$ è la **funzione di densità di probabilità** così definita:

$$f(t) = \lim_{\Delta t \rightarrow 0} \Pr(t \leq T < t + \Delta t) / \Delta t \quad e \quad \int_0^{\infty} f(u) d(u) = 1$$

cioè $f(t) \Delta t$ è la probabilità che l'evento avvenga tra t e $t+\Delta t$.

La **funzione rischio istantaneo** è invece [ARM71]:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \Pr\{(t \leq T < t + \Delta t) | T \geq t\} / \Delta t$$

$\lambda(t) \Delta t$ è la probabilità che l'evento avvenga tra t e $t+\Delta t$ condizionata al fatto di essere sopravvissuti fino a t , $\lambda(t)$ è un tasso di evento istantaneo.

Il **rischio istantaneo cumulativo** viene definito:

$$\Lambda(t) = \int_0^t \lambda(u) d(u)$$

La relazione tra S , λ ed f è:

$$\lambda(t) = f(t) / S(t) = -d \log S(t) / dt$$

da cui

$$\Lambda(t) = -\log S(t) \quad \text{o} \quad S(t) = \exp[-\Lambda(t)]$$

1.2.3 I metodi

La probabilità di sopravvivenza potrebbe essere stimata calcolando il rapporto tra il numero dei casi che hanno sperimentato l'evento e il totale dei soggetti inclusi nello studio (**metodo diretto**).

Ciò però implica che si realizzino contemporaneamente le seguenti condizioni: tutti i pazienti devono essere seguiti per un periodo maggiore o uguale a quello indagato; il rischio dell'evento terminale deve essere indipendente dalla durata dell'osservazione; l'insieme dei pazienti deve essere omogeneo per quanto riguarda la distribuzione dei tempi per l'evento terminale.

Il metodo diretto risulta quindi non adeguato, in quanto metterebbe a confronto pazienti nei quali l'evento è avvenuto dopo molto tempo dall'inizio dell'osservazione con altri seguiti per un periodo molto breve, nei quali l'evento potrebbe non essersi ancora verificato [POC83].

Il metodo corretto è invece rappresentato dalle tavole di sopravvivenza, dove la probabilità dell'evento viene calcolata in funzione del tempo.

Questo approccio consente di confrontare pazienti entrati nello studio in momenti diversi, che hanno differenti tempi di osservazione e che possono non avere sperimentato l'evento nel momento in cui si esegue l'analisi (censorizzati). Per effettuare l'analisi della sopravvivenza si utilizzano per lo più metodi non parametrici, non essendo nota la distribuzione teorica della sopravvivenza che si va ad analizzare, e questi sono essenzialmente due: metodo attuariale (Cutler-Ederer) [CUT58] e metodo del prodotto limite (Kaplan-Meier) [KAP58].

Il **metodo attuariale** ha tre caratteristiche essenziali: il tempo di osservazione è suddiviso a priori in una serie di intervalli generalmente di uguale ampiezza; assume che per i dati troncati, cioè in cui l'osservazione si ferma prima della fine dell'intervallo di tempo considerato, l'esperienza di vita sia simile a quella dei soggetti in osservazione; il rischio dell'evento non è considerato costante per tutto il periodo di osservazione, ma varia da intervallo ad intervallo [CUT58].

Il **metodo del prodotto limite** presenta le seguenti caratteristiche essenziali: non implica la suddivisione dei tempi in intervalli di ampiezza prefissata; la probabilità dell'evento è stimata in corrispondenza dei tempi in cui si verifica almeno un evento; il numero di eventi ed il numero di soggetti a rischio sono definiti solo in corrispondenza

dei tempi di risposta, e non dei tempi troncati; la probabilità di non avere l'evento rimane costante nell'intervallo di tempo tra due eventi; la curva che si ottiene è una funzione a gradini ed i tempi troncati influiscono sull'altezza di questi.

Questo metodo differisce nei risultati dal precedente in quanto effettua una sottostima iniziale ed una sovrastima finale della probabilità di sopravvivenza, determinando inoltre più ampi limiti di confidenza della stima, ma è ritenuto comunque meno rischioso essenzialmente perché non necessita di un'artificiosa suddivisione aprioristica della variabile tempo in intervalli non ideali per mettere in evidenza ciò che è avvenuto nella realtà. La probabilità condizionata di sopravvivenza viene stimata in corrispondenza di ciascuno dei tempi in cui si verifica almeno un evento terminale [KAP58].

1.2.4 Il metodo del prodotto limite (Kaplan-Meier)

Esso stima la curva di sopravvivenza in base al criterio di massima verosimiglianza.

Gli elementi che costituiscono la tavola di sopravvivenza sono:

$t_{(j)}$ = tempi di risposta (ordinati in modo crescente)

$d_{(j)}$ = numero di soggetti che presentano l'evento terminale al tempo $t_{(j)}$

$n_{(j)}$ = numero di soggetti esposti a rischio al tempo $t_{(j)}$

Il numero $n_{(j)}$ è formato da tutti i soggetti che sono senza evento terminale e sotto osservazione appena prima di $t_{(j)}$.

Se non vi sono evidenti ragioni contrarie, si assume che l'esperienza di vita dei soggetti persi di vista o usciti vivi alla fine dello studio sia la stessa di coloro che forniscono la risposta durante lo studio e quindi essi sono inclusi nell'insieme degli esposti a rischio sino a che sono sotto osservazione.

Le quantità $d_{(j)}$ e $n_{(j)}$ sono definite solo in corrispondenza dei tempi di risposta e non dei tempi troncati. Se un tempo troncato e uno di risposta sono uguali si considera, per convenzione, che il tempo troncato sia di almeno un infinitesimo più grande di $t_{(j)}$. È infatti verosimile e ragionevole assumere che il soggetto vivo al tempo troncato $t_{(j)}$ vivrà oltre e di conseguenza questi viene incluso tra gli esposti a rischio al tempo $t_{(j)}$.

Una stima della probabilità condizionata $q_{(j)}$ di presentare l'evento terminale all'istante $t_{(j)}$, dato che un soggetto è libero da evento fino all'istante immediatamente precedente, è data da:

$$q_{(j)} = \frac{d_{(j)}}{n_{(j)}} \quad j = 1, 2, \dots, J$$

e quindi la probabilità condizionata $p_{(j)}$ di non avere l'evento terminale al tempo $t_{(j)}$ è stimata da:

$$q_{(j)} = 1 - p_{(j)} = \frac{n_{(j)} - d_{(j)}}{n_{(j)}}$$

Negli "istanti" in cui non si rileva alcun evento terminale tale stima è pari a 1.

La stima della probabilità cumulativa di non avere l'evento terminale oltre l'istante $t_{(j)}$ è data da:

$$S_{(j)} = p_{(1)} * \dots * p_{(j)} = S_{(j-1)} * p_{(j)}$$

In ognuno dei punti $t_{(j)}$ dell'asse dei tempi la curva di sopravvivenza varia di un fattore $(1 - d_{(j)}/n_{(j)})$. Più precisamente, essendo $S_{(j)}$ la stima della probabilità di non avere l'evento terminale $t_{(j)}$, tale variazione avviene appena dopo l'istante $t_{(j)}$: la curva di sopravvivenza è dunque una funzione a gradini, che in corrispondenza dei tempi di risposta presenta punti di discontinuità (in essa è continua solo da sinistra). La curva non varia in corrispondenza dei tempi troncati ma la presenza di quest'ultimi influisce sull'altezza dei "salti" nei punti di discontinuità poiché interviene nel solo calcolo di $n_{(j)}$. Il calcolo della probabilità cumulativa di sopravvivenza è necessariamente limitato al tempo massimo di risposta [KAP58].

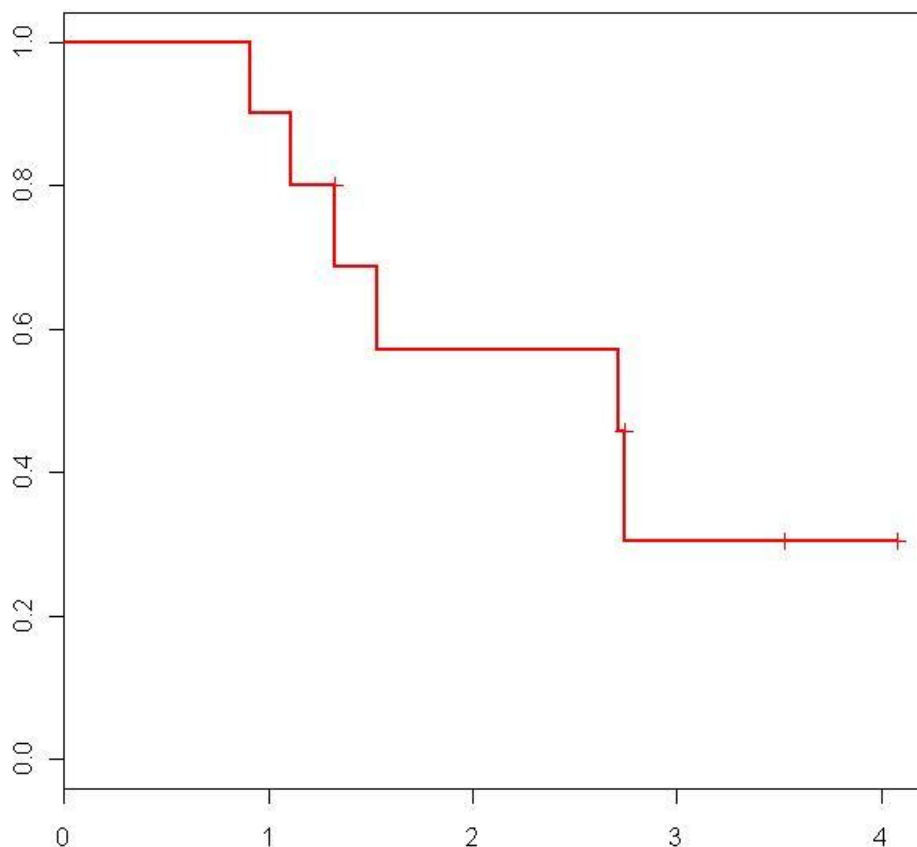


Figura 1.1 - Esempio di Curva di Kaplan Meyer

1.2.5 Il confronto di due curve di sopravvivenza (logrank test)

Quando in una sperimentazione clinica si decide di utilizzare la sopravvivenza per confrontare l'efficacia di un trattamento sperimentale e quella di un trattamento di controllo, occorre ricorrere all'impiego di opportuni metodi statistici.

Generalmente si utilizzano test non parametrici, quali il Wilcoxon e il Logrank (Mantel-Haenszel), in quanto è estremamente difficile avere a priori conoscenze sufficienti per ipotizzare la forma delle sottostanti funzioni teoriche di sopravvivenza [LAW82].

Il primo test risulta essere più sensibile alle differenze precoci tra le due curve di sopravvivenza confrontate, mentre il secondo è più sensibile a quelle tardive [PEA66].

Il logrank test è generalmente preferibile in quanto è il metodo più appropriato quando è soddisfatto l'assunto di proporzionalità dei rischi tra i due gruppi ed è ritenuto il più sensibile, tra i due, a cogliere differenze globali nelle curve di sopravvivenza.

Questo metodo confronta gli eventi osservati sotto ciascun trattamento con quelli attesi, qualora i trattamenti fossero in realtà parimenti efficaci.

Il risultato finale che si ottiene è un valore di chi-quadrato, da cui è possibile, mediante le tavole dei percentili della sua distribuzione, determinare il livello di significatività e per misurare la grandezza dell'eventuale differenza riscontrata tra i due trattamenti si potrà calcolare il rischio relativo di evento di un trattamento rispetto all'altro.

L'ipotesi attinente all'efficacia dei due trattamenti si esprime usualmente in termini di rischio istantaneo di morte (o di malattia) [BER81].

Indicando con H_0 ed H_1 rispettivamente l'ipotesi nulla e l'ipotesi alternativa si ha:

$$\begin{aligned}
 H_0: & \quad \lambda_A(t) = \lambda_B(t) & \quad \theta = 1 \\
 H_1: & \quad \lambda_A(t) = \theta \lambda_B(t) & \quad \theta \text{ diverso da } 1
 \end{aligned}$$

dove θ è l'ignota costante di proporzionalità.

Quando $\theta < 1$ il trattamento A è più efficace di B mentre è vero il contrario quando $\theta > 1$.

L'assunto per il test è quindi che il rapporto tra i rischi istantanei di morte nei due trattamenti sia costante nel tempo.

Essendo $N = N_A + N_B$ l'insieme dei pazienti entrati nello studio, siano $t_{(1)} < t_{(2)} < \dots < t_{(j)}$ i tempi di risposta ordinati in modo crescente indipendentemente dal trattamento.

Le informazioni riguardanti il tempo $t_{(j)}$ si possono riassumere in una tabella di contingenza come segue:

Trattamento	Morti (j)	Vivi (j)	A rischio (j)
A	$d_{A(j)}$	$n_{A(j)} - d_{A(j)}$	$n_{A(j)}$
B	$d_{B(j)}$	$n_{B(j)} - d_{B(j)}$	$n_{B(j)}$
Totale	d_j	$n_j - d_j$	n_j

Tabella 1.1- Tabella di Contingenza

Benché d_j , $n_{A(j)}$ e $n_{B(j)}$ siano variabili casuali dipendenti dalla esperienza precedente di sopravvivenza e dal processo di troncamento, il test di significatività può essere costruito solo assumendole fisse ad ogni tempo j .

Ciò comporta che uno solo dei valori nelle quattro celle della tabella sia una variabile casuale la cui funzione di probabilità può essere definita condizionatamente ai totali marginali osservati.

Per convenzione si considera come variabile casuale il numero di morti nel gruppo A la cui funzione di probabilità si dimostra essere l'ipergeometrica.

Sotto H_0 il valore atteso $E(d_{A(j)})$ di $d_{A(j)}$ è:

$$E(d_{A(j)}) = n_{A(j)} * \frac{d_{(j)}}{n_{(j)}}$$

e la sua varianza è:

$$\text{var}(d_{A(j)}) = \frac{n_{A(j)} * n_{B(j)} * d_{(j)} * (n_{(j)} - d_{(j)})}{n_{(j)}^2 * (n_{(j)} - 1)}$$

Il **logrank test** è, quindi, così definito:

$$Q_{M-H} = \frac{\{ \sum_{j=1}^J [d_{A(j)} - E(d_{A(j)})] \}^2}{\sum_{j=1}^J \text{var}(d_{A(j)})}$$

Questa statistica test si distribuisce asintoticamente, sotto l'ipotesi H_0 , come un χ^2_k , dove k è il numero di gruppi da confrontare meno uno.

La stima della costante di proporzionalità θ secondo Mantel e Haenszel è fornita dall'espressione:

$$\theta_{M-H} = \frac{\sum_{j=1}^J d_{A(j)} * (n_{B(j)} - d_{B(j)}) / n_j}{d_{B(j)} * (n_{A(j)} - d_{A(j)}) / n_j}$$

I limiti di confidenza esatti di θ_{M-H} presentano notevoli difficoltà di calcolo.

Si propone quindi di utilizzare una semplice procedura proposta da Miettinen [MIE76]:

$$\theta_{M-H} = \exp [\log \theta_{M-H} (1 \pm z_{1-\alpha/2} / Q_{M-H}^{1/2})]$$

1.3 Sistemi Informatici Radioterapia IRST

La radioterapia IRST dispone di diversi sistemi informatici che vanno dalla gestione della cartella clinica amministrativa del paziente fino al controllo dei macchinari presenti per la somministrazione della terapia.

Per questo progetto prendiamo in considerazione i software che hanno immagazzinati al loro interno i dati anagrafici e clinici dei pazienti al fine di effettuare analisi statistiche, valutando la risposta avuta in base alla malattia diagnosticata e il tipo di trattamento somministrato.

1.3.1 Cartella clinica ospedaliera

Log80 è il software che governa tutto il percorso del paziente all'interno della struttura, nel suo database sono presenti tutti i dati relativi al paziente ed è integrato alla maggior parte dei sistemi terzi presenti in struttura per il reperimento di informazioni specifiche. Essendo la sua architettura basata su una infrastruttura di tipo LAMP [LOG12] e cioè con server Linux su cui sono installati i servizi web di Apache e di database MySQL ed essendo programmato interamente in PHP è particolarmente versatile per quanto riguarda la diffusione delle informazioni e dei servizi offerti sulla rete ospedaliera dell'AUSL di Forlì che allo stato attuale conta diverse sedi distaccate, come ad esempio il reparto di oncologia e radioterapia che si trovano proprio all'IRST, ma anche alla condivisione delle informazioni con altre AUSL, per quanto riguarda l'oncologia il sistema è allo stato attuale utilizzato oltre che a Forlì anche a Cesena ed è in progetto l'inserimento dei reparti di Ravenna.

Grazie alle integrazioni con le anagrafiche ospedaliere/comunali per i comuni di Forlì e Cesena, sono reperibili tutti i dati anagrafici aggiornati dei pazienti in cura presso l'istituto e di tutti quelli che lo sono stati in passato.

Nel software e quindi all'interno del suo database, viene tracciato tutto il percorso del paziente all'interno dell'ospedale: gli accessi, le diagnosi, i referti, le prescrizioni, i medicinali, le terapie somministrate e molto altro.

Nel caso specifico della radioterapia i dati clinici come la diagnosi e la terapia scelta vengono però gestiti da un software terzo di cui si tratterà nel prossimo paragrafo.

Alla cartella clinica ospedaliera rimangono in ogni caso diversi compiti: mantiene i dati anagrafici aggiornati del paziente e tramite integrazione li trasferisce al software specifico di reparto; gestisce gli appuntamenti per le visite che si svolgono per determinare se sia necessario procedere con una terapia e le visite di controllo che si svolgono a cadenza regolare negli anni a seguire della stessa; registra tutti gli accessi del paziente in reparto e mediante integrazione importa tutti i dati relativi alle somministrazioni effettuate coi macchinari radioterapici; grazie a questa integrazione è in grado di generare stampe personalizzate per il paziente come referti o certificati di vario genere; sui dati importati è in grado di eseguire una codifica necessaria [LOG12], utile alla gestione del flusso amministrativo e l'esportazione dello stesso nel formato richiesto dalla regione per il rimborso delle prestazioni effettuate [IRS12b].

Oltre a questo, sempre grazie all'integrazione tra i due sistemi, per motivi di visibilità agli altri reparti, vengono importati in un secondo momento, tutti i dati clinici che la direzione sanitaria ha ritenuto opportuno rendere disponibili all'esterno del reparto di Radioterapia e Oncologia in carico all'IRST.

1.3.2 Cartella clinica di reparto

Il software MOSAIQ è un sistema in grado di centralizzare tutti i dati paziente di interesse per la radioterapia in un'unica interfaccia, accessibile da gruppi di utenti con competenze diverse dalle proprie postazioni all'interno della struttura ospedaliera.

Il software è basato su un'architettura client-server Windows e i dati sono immagazzinati e storicizzati su un database SQL Server di Microsoft [ELE10b].

Il sistema, leader nel settore, fornisce una serie di contenuti molto utili ai clinici, oltre ad un sistema di controllo e validazione delle dosi somministrate dagli acceleratori lineari (record & verify) ed è per questo che risulta essere insostituibile come software di reparto [ELE10b].

Il software offre una gestione dei dati in maniera strutturata, tramite un'interfaccia molto semplice da utilizzare sia per il personale infermieristico che per il personale medico.

Per quanto riguarda la parte amministrativa oltre la già citata integrazione per l'importazione e sincronizzazione delle anagrafiche, è presente una sezione denominata *Location* [ELE10b] dove è possibile creare e avere una gestione avanzata per le agende degli ambulatori e delle macchine con la possibilità, ad esempio, di innescare una serie di automatismi al momento della prenotazione: dal più semplice che prevede la stampa di un promemoria da consegnare al paziente, alla possibilità di associare all'appuntamento materiale di consumo che verrà scaricato dal magazzino di reparto e codici di prestazioni effettuate che verranno validate al momento dell'effettiva esecuzione della visita/trattamento, fino alla possibilità di esportare informazioni verso altri sistemi.

Nella cartella del paziente oltre ad annotazioni testuali è possibile anche raccogliere documenti di varia natura (file doc, pdf, e immagini) e in particolare immagini cliniche in formato DICOM [ELE10b], rappresentanti le varie fasi della preparazione del piano di cura, che grazie alla particolare struttura vengono riconosciute e indicizzate automaticamente dal sistema.

Andando in ordine cronologico rispetto al percorso del paziente in radioterapia a seguito degli appuntamenti viene eseguita una prima visita a cui segue una diagnosi [IRS12b].

All'interno del software sono presenti librerie specialistiche per il reparto di oncologia e radioterapia secondo gli standard internazionali come ad esempio la codifica ICD-O [ELE10b] rilasciate dall'organizzazione mondiale della sanità, che permette di catalogare le varie patologie in maniera univoca e riconosciuta internazionalmente.

A seguire vengono pianificate le TC di centratura le cui immagini serviranno per individuare la zona da curare e successivamente per verificare la posizione del paziente al momento della terapia, tali immagini vengono conservate all'interno del software e sono sempre disponibili per gli utilizzatori del programma a testimonianza della storia clinica del paziente.

Successivamente alla terapia vera e propria vengono effettuate, generalmente con cadenza semestrale delle visite di controllo (visite di follow-up) che servono a verificare lo stato di salute del paziente e a fini di ricerca statistica per valutare la risposta a un particolare trattamento eseguito [IRS12a].

1.4 Estrazione e presentazione dei dati clinici e amministrativi

Lo scopo del progetto è quello di realizzare un portale web in grado di presentare un'analisi dell'attività svolta in radioterapia secondo diversi punti di vista, utili al management dell'ospedale per valutare i risultati prodotti dal reparto ma anche, e soprattutto, da un punto di vista clinico per poter valutare gli effetti delle cure ai pazienti e migliorare dove possibile.

Per fare questo è necessario raccogliere una serie di dati dai vari sistemi utilizzati, elaborarli e presentarli all'interno di una interfaccia facilmente fruibile da tutti gli addetti al servizio.

Le richieste iniziali da parte della direzione sono state quelle di avere un quadro generale delle prestazioni svolte dalla radioterapia considerando diversi fattori quali: il periodo dell'anno, la provenienza del paziente e la tipologia di prestazione effettuata.

In oltre si è interessati a mostrare il livello di utilizzo degli ambulatori e delle macchine in grado di somministrare le terapie al fine di raccogliere statistiche utili per eventuali valutazioni e modifiche al servizio offerto al cittadino.

Per la parte amministrativa e conseguentemente economica il sistema di riferimento da cui estrarre le informazioni è Log80, software gestionale principale di tutto l'ospedale.

Per la parte che riguarda invece gli ambulatori e le macchine il software su cui vengono caricati i dati è MOSAIQ e sarà quindi il suo database a fornire le informazioni necessarie.

Oltre alle richieste della direzione IRST, il responsabile clinico del reparto ha chiesto di valutare la possibilità di estrarre in tempo reale la situazione dei pazienti in *follow-up* effettuando valutazioni di tipo statistico secondo lo standard adottato nella ricerca clinica delle curve di sopravvivenza (kaplan-meier) al verificarsi di diversi eventi.

Sono stati scelti indicatori per la sopravvivenza generale del paziente (*Overall Survival o OS*), un indicatore che valutasse il tempo prima di un'eventuale ricaduta della malattia (*Disease-free Survival o DFS*) e uno nel caso in cui il trattamento non avesse portato a una guarigione completa, per valutare il tempo nel quale la malattia ricominci eventualmente a progredire dopo la terapia (*Progression-free Survival o PFS*) [CLI11].

In questo caso specifico è stato necessario valutare assieme ai medici il modo in cui la cartella di reparto veniva compilata ed è stato fondamentale modificare il metodo andando a individuare i campi in cui poter inserire le informazioni utili alla statistica.

E' stato necessario rivalutare il metodo e la completezza di inserimento delle diagnosi inserite e delle tabelle di follow-up, in modo da poter raggruppare i pazienti secondo diversi fattori:

- I dati anagrafici utili all'individuazione del paziente come: il codice di reparto, codice fiscale, nome, cognome, data di nascita, sesso, data morte ;
- I dati di diagnosi: data, descrizione della diagnosi, codice ICD-O, area di origine della malattia, categoria, lateralità, descrizione della istologia, codice dell'istologia e la classificazione dello stadio della malattia attraverso il sistema di codifica internazionale TNM;
- I dati relativi al follow-up del paziente come: data di visita, indicazione di una modifica del TNM secondo dei parametri codificati per indicare una progressione o regressione della malattia, e altri dati utili a rilevare la salute del paziente connessi non alla malattia ma alla terapia subita.

CAPITOLO 2.

PROGETTAZIONE E TECNOLOGIE UTILIZZATE

2.1. Analisi dei software e dei database presenti

In questo paragrafo si analizzeranno i software partendo dalle schermate utilizzate dal personale di reparto in modo da comprendere al meglio il tipo di interfaccia e le facilitazioni che essa propone al personale di radioterapia.

Questo servirà poi, eventualmente, per riproporli in maniera simile all'interno del software che si andrà a sviluppare.

Altro motivo di analisi visiva dei software, data la mancanza di documentazione in merito, è per capire l'organizzazione dei dati in modo poi da facilitare l'analisi del database in cui questi sono inseriti.

2.1.1 Software Amministrativo Ospedaliero

Il software è LOG80, è un applicativo sviluppato con tecnologie web e più precisamente si appoggia alla piattaforma LAMP (Linux, Apache, PHP e MySQL), le schermate di cui possiamo vedere qualche esempio a seguire sono disegnate in maniera elementare, presentano però una grande quantità di dati che solitamente distrae l'utilizzatore non esperto.

Tuttavia il software permette una grande flessibilità nella gestione dei flussi e una serie di integrazioni che lo rendono insostituibile, almeno per il momento, all'interno dell'ospedale.

Per la parte che ci interessa andiamo ad analizzare, le agende e le cartelle ambulatoriali del singolo paziente, in modo da capire se all'interno del programma siano presenti o meno i dati necessari all'analisi richiesta.

Le agende LOG80 rappresentano la programmazione dei singoli ambulatori, sono configurabili in modo da assegnare per ogni giorno della settimana in un determinato ambulatorio un numero di slot disponibili, e in ogni slot è possibile associare una prestazione predefinita per facilitare l'inserimento dell'appuntamento da parte del personale addetto all'accettazione pazienti.

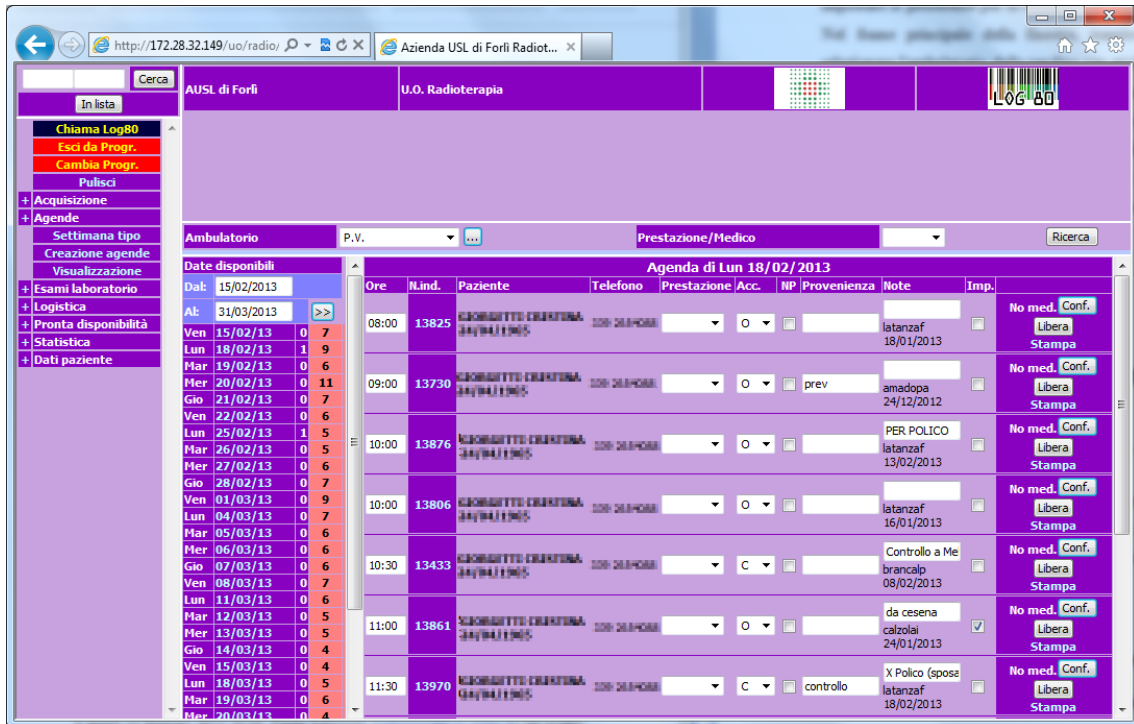


Figura 2.1 - Schermata agende LOG80

Come possiamo vedere nella *figura 2.1*, sulla parte sinistra troviamo un menù ad albero nel quale è presente la sezione agende, entrando con il profilo di amministratore del reparto sotto questa tendina possiamo vedere oltre che visualizzazione, disponibile a tutto il personale, anche i pulsanti "settimana tipo" e "creazione agenda" questi due pulsanti non sono utili al nostro scopo ma servono come detto in precedenza a impostare le preferenze per la creazione di una nuova agenda (o nuovo ambulatorio).

Nel frame principale della finestra, tramite il menu nella parte alta, è possibile selezionare l'ambulatorio dalla tendina con etichetta omonima.

Subito sotto verranno visualizzati sulla sinistra l'elenco dei giorni in cui l'ambulatorio è aperto, il numero di *slot* disponibili e il numero totale di *slot*.

Nella parte centrale troviamo invece i singoli appuntamenti dove vengono indicati i principali dati anagrafici, l'orario dell'appuntamento e la prestazione prevista per il tipo di ambulatorio oltre che un piccolo campo note utilizzato come promemoria per il personale.

PINCO PALLA
 01/01/1900 - età 113 - sesso M
 FORLI' (FC)
 Cod. az. 000000000H0357734
 USL 080111 FORLI'

Note: []

Conferma Certif. Patologie Stampa copertina

Ricerca in A.A. Per segnalare errori sull'anagrafica e/o l'impossibilità di modificare i dati:

ACCESSI/CARTELLE						
Stampa	N. interno	T.E.	Data entrata	Data uscita	U.O.	Mod. accesso
1	*** ANNULLATA 006427_20110215 da ***	RIC	15/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	Accesso diretto
2	*** ANNULLATA 006427_20110214 da ***	ERRORE	14/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	
3	Ref. Etic. Etic.G 006427_20110214 R Ripeti	T	14/02/2011 00:00:00	00/00/0000 00:00	RADIOTERAPIA	Accesso diretto
4	Ref. Etic. Etic.G 00724/11	T	11/02/2011 02:24:00	00/00/0000 00:00	RADIOTERAPIA	Accesso diretto
5	*** ANNULLATA 00614/11 da ***	T	08/02/2011 13:14:00	00/00/0000 00:00	RADIOTERAPIA	Accesso diretto
6	*** ANNULLATA 006427_20110208 da ***	ERRORE	08/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	
7	*** ANNULLATA 006427_20110208 da ***	RIC	08/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	Accesso diretto
8	*** ANNULLATA 006427_20110208 da ***	RIC	08/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	Accesso diretto
9	*** ANNULLATA 006427_20110208 da ***	RIC	08/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	Accesso diretto
10	*** ANNULLATA 006427_20110208 da ***	RIC	08/02/2011 00:00:00	00/00/0000	RADIOTERAPIA	Accesso diretto
11	*** ANNULLATA 03117/10 da ***		05/07/2010 10:07:00		RADIOTERAPIA	Interno
12	*** ANNULLATA 006427_20100523 da ***	ERRORE	23/05/2010 00:00:00	00/00/0000	RADIOTERAPIA	
13	*** ANNULLATA 006427_20100523 da ***	ERRORE	23/05/2010 00:00:00	00/00/0000	RADIOTERAPIA	
14	*** ANNULLATA 006427_20100523 da ***	ERRORE	23/05/2010 00:00:00	00/00/0000	RADIOTERAPIA	
15	Ref. Etic. Etic.G 00216010	PV	09/03/2010 16:20:00	00/00/0000 00:00	RADIOTERAPIA	Interno
16	*** ANNULLATA 00216010 da ***		01/03/2010 15:52:00		RADIOTERAPIA	Interno
17	Ref. Etic. Etic.G 00216010	PV	05/02/2010 12:31:00	00/00/0000 00:00	RADIOTERAPIA	Controllo
18	*** ANNULLATA 006427_20100128 da ***	ERRORE	28/01/2010 00:00:00	00/00/0000	RADIOTERAPIA	
19	*** ANNULLATA 6427_2010 da ***	PV	27/01/2010 12:13:00	00/00/0000	RADIOTERAPIA	
20	*** ANNULLATA 006427_20100127 da ***	ERRORE	27/01/2010 00:00:00	00/00/0000	RADIOTERAPIA	

Figura 2.2 - Schermata accessi paziente LOG80

Cliccando sull'identificativo numerico accanto al paziente è possibile aprire la schermata relativa al paziente e visualizzare una serie di informazioni relative agli accessi dello stesso in reparto come mostrato in figura 2.2.

Andando poi a cliccare sulle singole righe, rappresentanti gli accessi del paziente, si ha una visualizzazione chiara delle prestazioni eseguite per quella determinata visita e la possibilità di stampare in vari formati referto e promemoria figura 2.3.

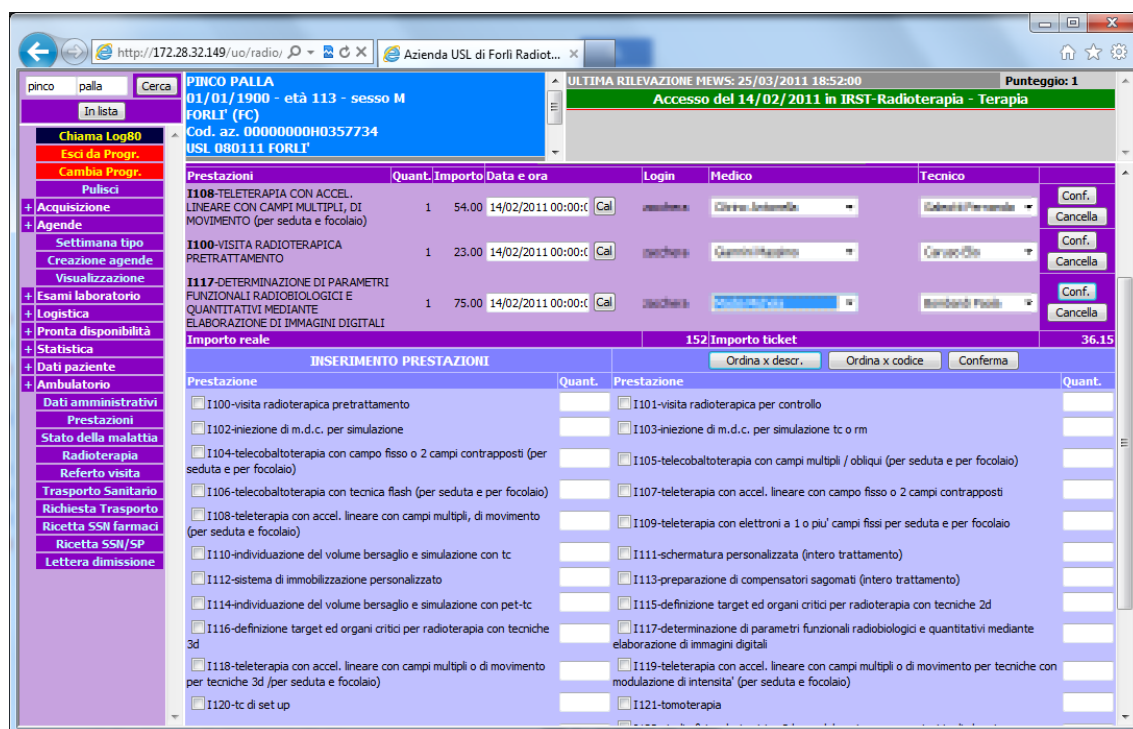


Figura 2.3 - Schermata registrazione e controllo prestazioni effettuate

Semplicemente analizzando le procedure [IRS12a] di accettazione del paziente e andando ad utilizzare il software per qualche minuto è emerso subito che non ci sia una correlazione diretta tra le prenotazioni e gli accessi effettivamente registrati per il paziente inoltre sono totalmente assenti i dati relativi all'occupazione delle macchine per le TC e la radioterapia, questo rende i dati inseriti in agenda LOG80 inaffidabili o quantomeno incompleti, per questo motivo si farà affidamento alla gestione degli appuntamenti del software di reparto per presentare i prospetti di occupazione presente, passata e futura.

Essendo il gestionale utilizzato principalmente per la gestione dei flussi amministrativi e dei rimborsi possiamo invece considerare ottimale l'estrazione di questi ultimi dal database LOG80.

In modo da presentare i dati aggregati relativi alla mole di lavoro svolto, come richiesto dalla direzione dell'ospedale.

I dati da presentare sono principalmente il numero di prestazioni divise per tipo provenienza e fascia di età del paziente.

Per il raggruppamento in base alla provenienza del paziente è stata richiesta: la visualizzazione dettagliata per AUSL di appartenenza all'interno di area vasta Romagna e cioè delle unità sanitarie di Forli, Cesena, Ravenna, Rimini e Imola; un

raggruppamento per tutte le AUSL di area vasta; e un raggruppamento per tutti gli accessi provenienti dalle aree al di fuori dell'area vasta Romagna.

2.1.2 Analisi database amministrativo ospedaliero

Andando ad analizzare il database con il supporto della software house che ha sviluppato il software sono state individuate le tabelle relative all'anagrafica dei pazienti, la registrazione delle prestazioni effettuate e quella degli accessi effettuati.

Da queste tabelle è stata generata una vista, poi resa direttamente accessibile al software in sviluppo, contenente tutte le informazioni utili alla rappresentazione grafica e tabellare richiesta.

Per comodità, vista anche la quantità di campi presenti nelle singole tabelle, verranno elencati i soli campi utili alla creazione della vista, le tabelle coinvolte sono:

- RADIO.PREST_REP
id_prest, *k_pri*, *c_prest*, *q_prest*, *i_prest*, *data_ora*, *login*, ...
- RADIO.CART_REP
k_pri, *n_ind*, *u_op*, *f_ann*, ...
- RADIO.ANA_REP
n_ind, *Sesso*, *d_nas*, *r_usl* ...
- DBAZIENDA.TBLPREST
c_rep, *c_prest*, *p_descr*, *p_val*, ...

La tabella "Radio.Prest_Rep" è la tabella dove sono contenute tutte le prestazioni registrate in reparto e tutti i dati relativi a esse, *id_prest* è chiave primaria della tabella, *k_pri* è l'identificatore esterno della tabella "Radio.Cart_Rep" e serve appunto per mettere in relazione le due tabelle, *c_prest* è il codice relativo alla prestazione registrata che può essere decodificato mettendo in relazione la tabella con "DbAzienda.TblPrest", *q_prest* è un valore numerico che indica la quantità di prestazioni di tipo *c_prest* effettuate, *i_prest* è il valore della singola prestazione al momento della registrazione, *data_ora* rappresenta l'ora e la data di inserimento della prestazione nel database e infine *login* il nome utente dell'operatore che ha registrato la prestazione (medico o infermiere).

Nella tabella "Radio.Cart_Rep" vengono registrati tutti gli accessi al reparto, qui troviamo oltre alla chiave primaria *k_pri* già vista nella tabella precedente anche il codice univoco del paziente *n_ind*, il codice dell'unità operativa di radioterapia *u_op* e il campo *f_ann* che indica se l'accesso del paziente è stato annullato o meno, in questo caso sarà necessario non considerare le prestazioni legate a questo particolare accesso.

Nella tabella "Radio.Ana_Rep" sono presenti i dati anagrafici dei pazienti, di questa tabella ci interessano principalmente il campo *n_ind* chiave primaria e identificativo del paziente, il campo *r_usl* contenente il codice della AUSL di provenienza del paziente, il campo *Sesso* e il campo *d_nas*, che indica la data di nascita del paziente, per poter raggruppare i pazienti per genere e fascia di età.

Infine dalla tabella "DbAzienda.TblPrest" tramite i campi *c_rep* e *c_prest* è possibile identificare il campo *p_descr* contenente una descrizione testuale della prestazione e *p_val* che indica il valore corrispettivo in denaro.

Query di creazione della vista:

```
CREATE VIEW irst.vista_prestazioni AS
SELECT
    p.id_prest,          //ID prestazione
    c.k_pri,            //ID cartella
    p.c_prest,         //codice prestazione
    p.q_prest,         //quantità prestazione
    p.i_prest,         //valore prestazione
    p.data_ora,        //data registrazione della prestazione
    p.login,           //login di registrazione della prestazione
    a.n_ind,           //ID paziente
    c.u_op,            //codice unità operativa
    c.f_ann,           //indice di cartella annullata
    a.Sesso,           //sesso del paziente
    a.d_nas,           //data nascita del paziente
    a.r_usl,           //AUSL di provenienza del paziente
    d.p_descr,         //Descrizione testuale prestazione
FROM radio.prest_rep AS p
JOIN radio.cart_rep AS c
    ON (prest_rep.k_pri=cart_rep.k_pri
        AND cart_rep.f_ann<>'x')
JOIN radio.ana_rep AS a
```



```

        ON (cart_rep.n_ind=ana_rep.n_ind)
JOIN dbazienda.tblPrest AS d
        ON (cart_rep.u_op=tblPrest.c_rep
            AND prest_rep.c_prest=tblPrest.c_prest)
WHERE
        cart_rep.u_op = "CODICE RADIOTERAPIA"
    
```

Rappresentazione grafica delle relazioni tra le tabelle utilizzate:

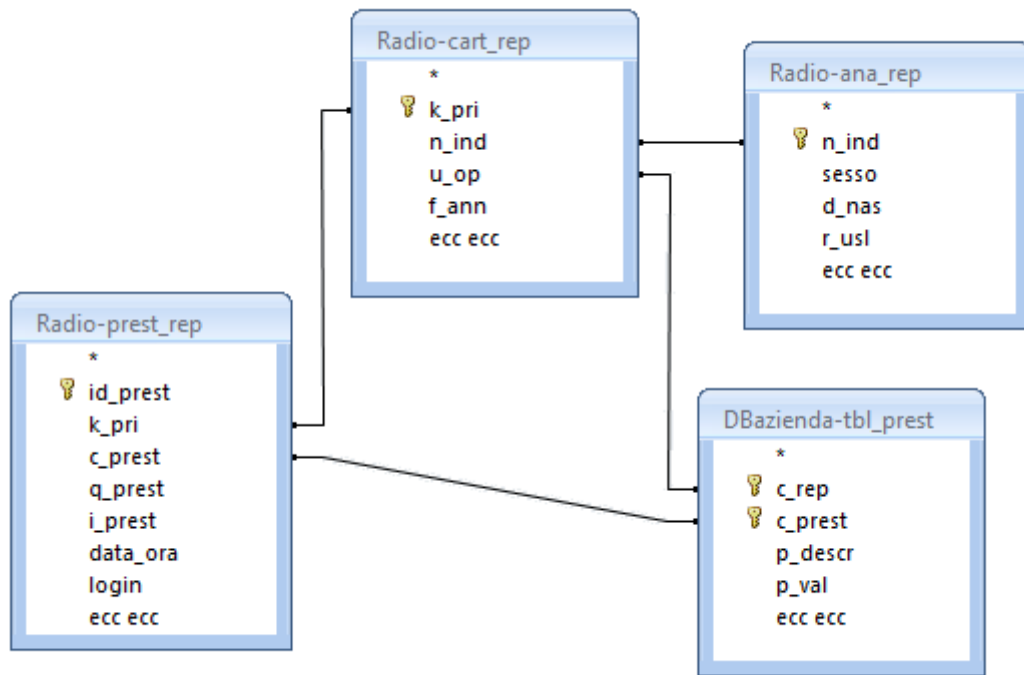


Figura 2.4 - Relazioni tabelle per vista LOG80

Mettendo in *JOIN* le tabelle secondo i criteri visibili in *figura 2.4* e ancor meglio specificati nel codice SQL antecedente è stato possibile creare una vista con tutti i dati utili alla rappresentazione grafica richiesta.

2.1.3 Software clinico e amministrativo radioterapia

Il software utilizzato per la gestione amministrativa ma soprattutto clinica del reparto di radioterapia è Mosaiq.

L'applicazione è basata una architettura *client-server* classica, il client è stato installato su tutti i computer presenti in radioterapia nelle sedi di Forlì e Meldola.

Il server è invece locato in un armadio dedicato all'interno della sala server dell'istituto IRST di Meldola, sulla macchina, con sistema operativo Windows Server 2003r2 è installato il software necessario al funzionamento dei vari client e il servizio SQL Server di Microsoft come motore per il database.

A causa di un problema legato sia al software di LOG80 che alla procedura precedentemente analizzata e di conseguenza non avendo potuto reperire le informazioni necessarie alla rappresentazione dello stato degli ambulatori, procediamo ad analizzare il sistema di prenotazione degli appuntamenti fornito con Mosaiq chiamato "Location".

Anche in questo software accedendo a schermate particolari l'operatore amministratore è in grado di creare o eliminare nuovi ambulatori o diagnostiche, impostare un tempo di apertura per ciascuno di essi all'interno di un calendario settimanale e impostare la durata prevista per ciascuna "visita" per il singolo ambulatorio.

Come per LOG80 è possibile impostare una prestazione predefinita per ogni location.

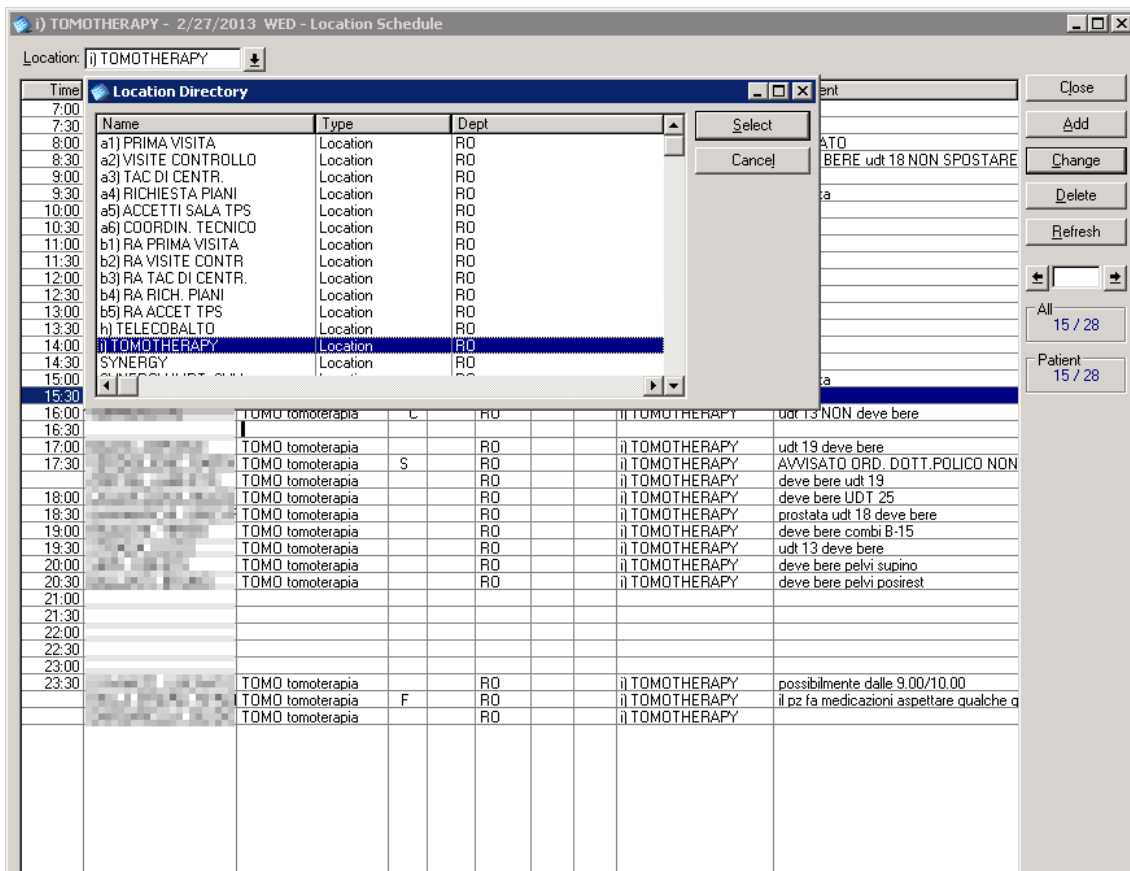


Figura 2.5 - Schermata Location Mosaiq

Per inserire un appuntamento l'operatore deve selezionare, o nel caso della prima volta importare da LOG80, i dati del paziente.

In questo modo il programma visualizza una serie di dati relativi al paziente e andando poi nella sezione location, raggiungibile tramite un'icona nella pagina principale, è possibile selezionare l'ambulatorio, scegliere la data e visualizzando il contenuto, si potrà assegnare l'orario di visita al paziente come mostrato in *figura 2.5*.

Per ogni appuntamento registrato l'operatore di accettazione può inoltre specificare delle note visibili al medico e stampare un promemoria da dare al paziente.

Quando il paziente si presenta effettivamente per la visita il medico, o nel caso di terapia il tecnico, deve partire proprio dalla schermata di location per accettare il paziente.

Questa procedura obbligata dal programma, ed eseguita dagli operatori, permette di avere una connessione logica dei dati del paziente dal momento dell'appuntamento al momento della registrazione delle prestazioni, e ci consente di aggregare i dati in modo da mostrare, su una linea temporale, quanti siano gli *slot* prenotati nel passato, nel presente ma anche nel futuro per singola *location*.

Per quanto riguarda le sedute di terapia oltre che agli appuntamenti assegnati, sarà poi possibile andare ad esaminare, per ogni singola prenotazione nel passato, quale sia stato l'esito, ovvero se la terapia si è effettivamente svolta o meno, e nel caso in cui non si sia svolta quali siano stati gli impedimenti causa della mancata erogazione di servizio (indisposizione del paziente, problema tecnico alla macchina).

Al momento della "cattura" della prenotazione è possibile impostare un esito diverso da quello dell'avvenuta somministrazione mediante l'utilizzo di un campo dedicato.

Time	Patient	Activity	Status	Dept	Staff	MD	Location	Comment
7:00								
7:30								
8:00		TOMO tomoterapia	SC	RO			ij TOMOTHERAPY	AVVISATO
8:30		TOMO tomoterapia	SC	RO			ij TOMOTHERAPY	AVVISATA
9:00								
9:30		TOMO tomoterapia	C	RO			ij TOMOTHERAPY	
10:00		TOMO tomoterapia	FC	RO			ij TOMOTHERAPY	
10:30		TOMO tomoterapia	B	RO			ij TOMOTHERAPY	
11:00		TOMO tomoterapia	C	RO			ij TOMOTHERAPY	
11:30		TOMO tomoterapia	SC	RO			ij TOMOTHERAPY	Avvisato
12:00		TOMO tomoterapia	C	RO			ij TOMOTHERAPY	
12:30		TOMO tomoterapia	C	RO			ij TOMOTHERAPY	
13:00		TOMO tomoterapia	SC	RO			ij TOMOTHERAPY	avvisato
13:30		TOMO tomoterapia	C	RO			ij TOMOTHERAPY	
14:00								
14:30								
15:00								
15:30								
16:00								
16:30								

Figura 2.6 - Esito della seduta di terapia Mosaicq

Successivamente sulla colonna status, come è possibile vedere in *figura 2.6*, vengono visualizzate le codifiche dell'esito della seduta: S indica un inizio di terapia (le terapie come detto in precedenza sono composte da diverse sedute), F l'ultima seduta di una terapia, C significa terapia eseguita (secondo il gergo Mosaiq "catturata"), B un'interruzione di terapia dovuto ad una indisposizione del paziente, M un fermo macchina, mentre la casella vuota significa che la terapia non è ancora stata somministrata o almeno non è ancora stato assegnato uno status ad essa.

Per la parte relativa alle curve di sopravvivenza invece i dati necessari alla creazione delle stesse vengono inseriti subito successivamente al momento della visita da parte del medico.

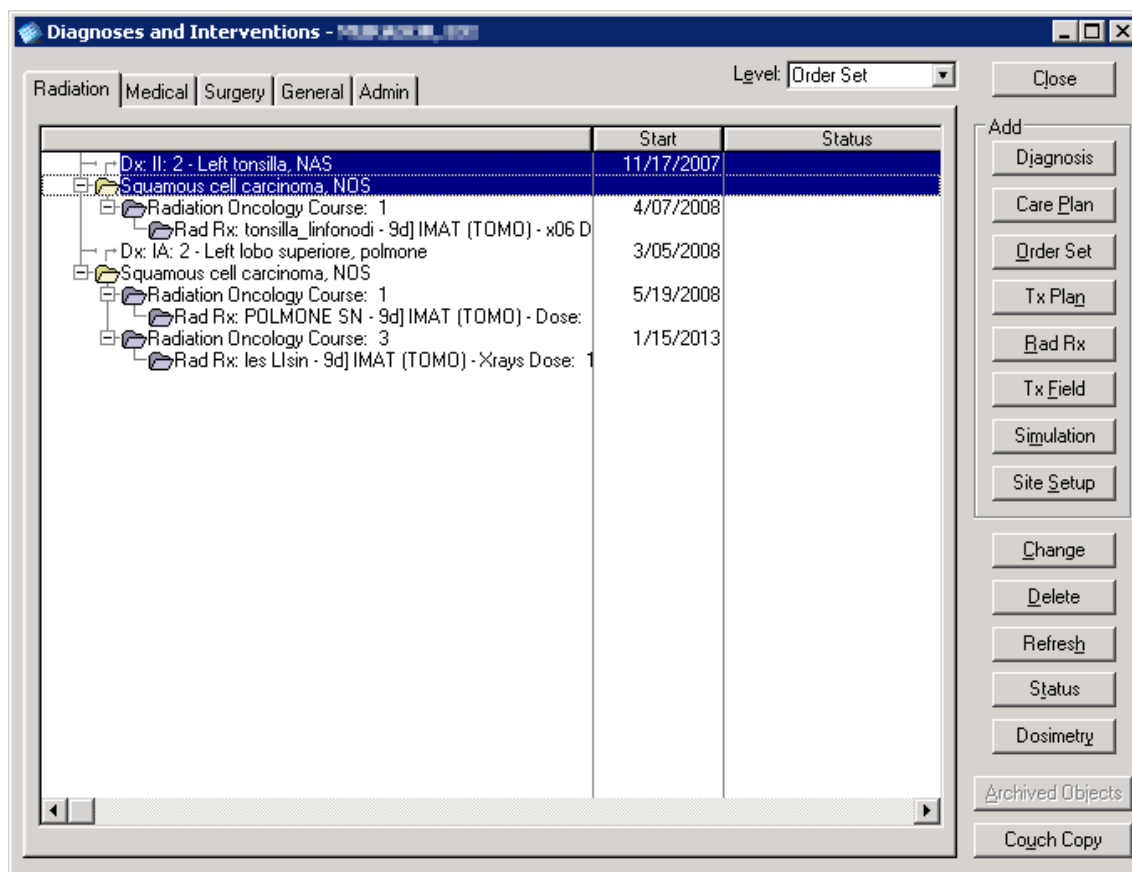


Figura 2.7 - Finestra Diagnoses and Interventions Mosaiq

In *figura 2.7* possiamo vedere la schermata riepilogativa delle diagnosi effettuate su un paziente alle quali conseguono poi trattamenti organizzati secondo una struttura ad albero di facile comprensione.

Facendo *click* sul pulsante "Diagnosis" è possibile inserire una nuova diagnosi per il paziente.

Nella schermata di compilazione dei dati della diagnosi (*figura 2.8*), che viene compilata dal medico nel momento immediatamente successivo alla prima visita, in sono presenti quasi tutte le informazioni per la parametrizzazione della ricerca per le curve di sopravvivenza, possiamo infatti trovare il codice e la descrizione della malattia, la stadiazione (*stage*) calcolata secondo i parametri T, N, e M, categoria della malattia, l'istologia (campo *morphology*) e la data della diagnosi.

Figura 2.8 - Diagnosis and Staging Mosaic

Gli unici campi non presenti all'interno di questa schermata sono il tipo di terapia effettuata sul paziente visibile nella *figura 2.7* e la data morte del paziente che è recuperabile dalla finestra presentata in *figura 2.9*, utilizzata per l'esito delle visite di controllo che vengono effettuate successivamente alla terapia (visite di *follow-up*).

Proprio nella schermata di follow-up si è ipotizzato un problema, poi verificato andando ad indagare sul database, che ci ha bloccati nello sviluppo di curve diverse da quella di *overall-survival*.

Come primo indizio si può notare nella schermata di compilazione dell'esito della visita di controllo che è possibile inserire la data morte (*expired*), in questo modo è possibile associare al paziente più date di morte, cosa che ovviamente non trova riscontro nella realtà, andando poi più a fondo nell'analisi della base di dati, si è notato che le visite di follow-up sono collegate logicamente al paziente e non alla diagnosi.

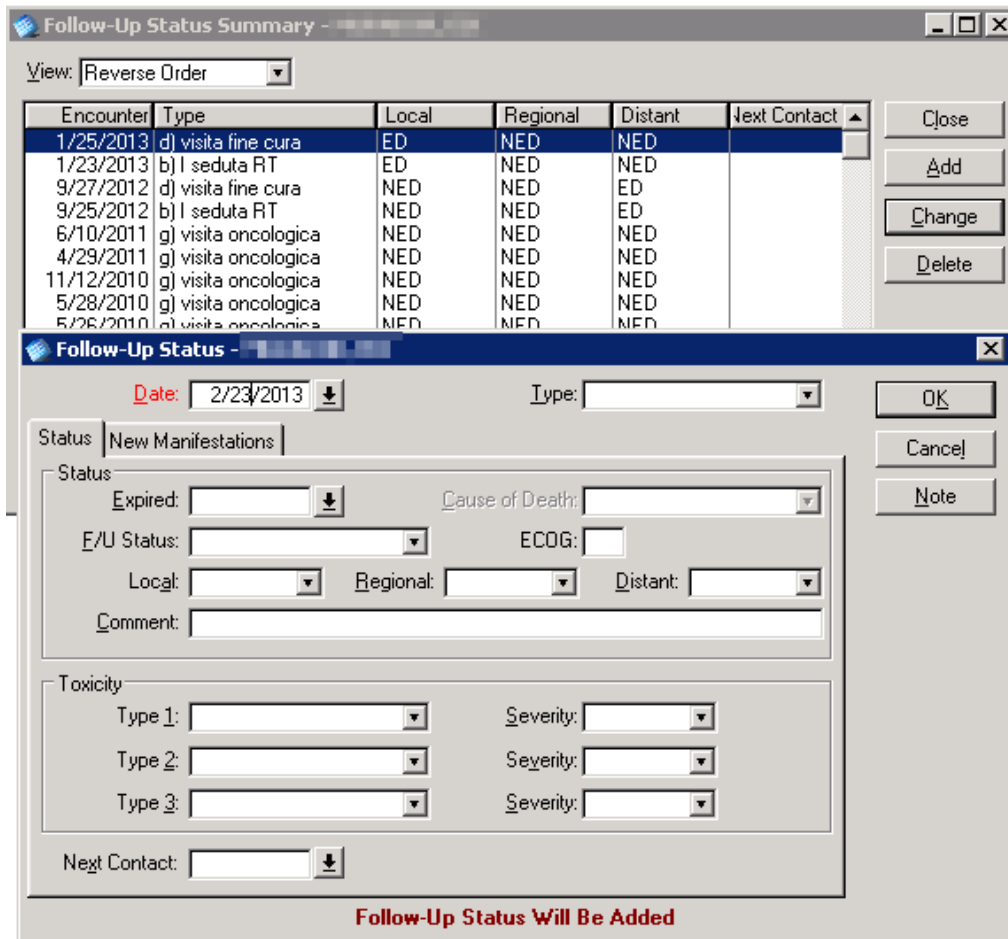


Figura 2.9 - Finestre Follow-Up summary e Follow-Up Mosaic

Questa caratteristica del software impedisce l'analisi dei dati relativi alle curve *disease-free survival*, e *progression-free survival* nel caso di patologie multiple, per le quali è necessaria un'analisi temporale dei campi *Local*, *Regional* e *Distant* in relazione alla data di diagnosi.

2.1.4 Analisi database Mosaic

Seguendo le indicazioni di un esperto della società Elekta Italia distributrice del prodotto Mosaic per la nostra nazione è stato possibile analizzare il database.

Mi è stato creato un accesso in sola lettura al database del software Mosaiq con il quale si è potuto analizzare la struttura delle tabelle e delle viste al suo interno.

Proprio tra le viste già presenti ne è stata identificata una contenente tutti i dati utili per l'analisi dell'occupazione degli ambulatori e delle diagnostiche (genericamente *location* all'interno di Mosaiq).

All'interno della vista `dbo.vw_Schedule` si trovano, tra gli altri, i seguenti campi:

- `Sch_Id`: identificativo di riga della vista
- `App_DtTm`: data e ora dell'appuntamento
- `Duration_time`: durata previsto per lo svolgimento della prestazione
- `Activity`: prestazione prevista per l'appuntamento
- `Last_Name`: cognome del paziente
- `First_Name`: nome del paziente
- `IDA`: identificativo univoco del paziente
- `Short_Desc`: nome della location
- `Location_ID`: identificativo numerico della location
- `account_status`: valori inseriti all'interno della colonna status vista in *figura 2.6*

A causa della lunghezza della *query* necessaria alla generazione della vista ci si limiterà ad elencare le tabelle e i vincoli di *JOIN* utilizzati per la generazione della stessa:

- `dbo.Schedule`: tabella contenente tutti i dati relativi alla prenotazione degli appuntamenti
- `dbo.Medical`: tabella con i dati delle diagnosi
- `dbo.Patient`: tabella con parte dei dati anagrafici dei pazienti
- `dbo.staff`: tabella con i dati anagrafici degli utenti del programma
- `dbo.Queue`: tabella per la storicizzazione delle modifiche apportate alle prenotazioni e non solo
- `dbo.CPT`: tabella contenente i dati relativi alle terapie prescritte
- `dbo.Config`: tabella contenente i dati relativi al reparto di radioterapia
- `dbo.Ident`: tabella per la storicizzazione delle modifiche apportate alle anagrafiche dei pazienti
- `dbo.Admin`: tabella con parte dei dati anagrafici dei pazienti

```

FROM
dbo.Schedule AS SCH WITH
LEFT OUTER JOIN
dbo.Medical AS MED WITH ON SCH.Pat_ID1 = MED.PAT_ID1
LEFT OUTER JOIN
dbo.Patient AS PAT WITH ON SCH.Pat_ID1 = PAT.Pat_ID1
LEFT OUTER JOIN
dbo.Staff AS STF WITH ON SCH.Staff_ID = STF.Staff_ID
LEFT OUTER JOIN
dbo.Queue AS QUE WITH ON SCH.Sch_Set_Id = QUE.Sch_ID
LEFT OUTER JOIN
dbo.CPT WITH ON SCH.Activity = dbo.CPT.Hsp_Code
LEFT OUTER JOIN
dbo.Config AS CFG WITH ON SCH.Inst_ID = CFG.Inst_ID
LEFT OUTER JOIN
dbo.Staff AS QLOC WITH ON QUE.Location = QLOC.Staff_ID
LEFT OUTER JOIN
dbo.Admin AS ADM WITH ON SCH.Pat_ID1 = ADM.Pat_ID1
LEFT OUTER JOIN
dbo.Staff AS ADMStaff WITH ON ADM.Attending_Md_Id = ADMStaff.Staff_ID
LEFT OUTER JOIN
dbo.Staff AS LOC WITH ON SCH.Location = LOC.Staff_ID
LEFT OUTER JOIN
dbo.Ident WITH ON SCH.Pat_ID1 = dbo.Ident.Pat_Id1
WHERE
(SCH.Version = 0)
AND (MED.MED_ID IS NULL)
AND (QUE.Version IS NULL OR QUE.Version = 0)
AND (dbo.Ident.Version IS NULL OR dbo.Ident.Version = 0)
OR (SCH.Version = 0)
AND (QUE.Version IS NULL OR QUE.Version = 0)
AND (dbo.Ident.Version IS NULL OR dbo.Ident.Version = 0)
AND (MED.Seq = 1)

```

Per le curve di sopravvivenza invece si è dovuto procedere andando a ricercare singolarmente i campi necessari all'interno delle tabelle del database e in un secondo

momento si è passati alla creazione di una vista ad hoc che riassume tutti i campi necessari.

Si fa notare che la creazione di questa vista è utilizzata anche per altri fini in software per il controllo della qualità dei dati, scollegati dal progetto presentato in questa tesi.

Di seguito un elenco dei campi disponibili all'interno della vista:

Alias	Descrizione	Tabella.Campo
Pat_ID1	Identificativo paziente	Admin.Pat_ID1
CC	Codice cartella	dbo.fn_GetPatDisplayId (dbo.Admin.Pat_ID1, dbo.fn_GetActiveDept(), 1)
CF	Codice fiscale	dbo.fn_GetPatDisplayId(dbo.Admin.Pat_ID1, dbo.fn_GetActiveDept(), 2)
Cognome	Cognome Paziente	Patient.Last_Name
Nome	Nome Paziente	Patient.First_Name
Inattivo	Stato anagrafica paziente	Patient.Inactive
Sesso	Sesso paziente	Admin.Gender
Data_di_Nascita	Data di nascita	Patient.Birth_DtTm
Telefono_casa	Telefono abitazione	Admin.Pat_Home_Phone
Cellulare	Telefono cellulare	Admin.Pat_CellPhone
Indirizzo	Indirizzo residenza	Admin.Pat_Adr1
Citta	Città di residenza	Admin.Pat_City
Regione	Regione di residenza	Admin.Pat_State
Nazione	Nazione di residenza	Admin.Pat_Country
Data_Diagnosi	Data della diagnosi	Medical.Dx_DtTm
Eta_alla_diagnosi	Età alla diagnosi	DATEDIFF(yyyy, dbo.Patient.Birth_DtTm, dbo.Medical.Dx_DtTm)
Diagnosi	Descrizione diagnosi	Topog.Description
Codice_Diagnosi	Codice diagnosi	Medical.Topography
Origine_Diagnosi	Origine della malattia	Medical.Category
Categoria	Categoria della diagnosi	Medical.Diagnosis_Class
Lateralita	Lateraltà diagnosi	Medical.Paired_Organ
Codice_Istologia	Codice istologia	Medical.Histology
Istologia	Descrizione istologia	Morphol.Description
T_Clinico	T clinico	TNMStage.T_Stage
N_Clinico	N clinico	TNMStage.N_Stage
M_Clinico	M clinico	TNMStage.M_Stage
Stadio_Clinico	Stadio clinico	TNMStage.Stage
T_Patologico	T patologico	TNMStage2.T_Stage
N_Patologico	N patologico	TNMStage2.N_Stage
M_Patologico	M patologico	TNMStage2.M_Stage

Alias	Descrizione	Tabella.Campo
Stadio_Patologico	Stadio patologico	TNMStage2.Stage
HP_Grade	Performance status	Medical.Hist_Grade
Data_Consulto	Data prima visita	Medical.Diag_DtTm
Data_Incontro	Data visita followup	Followup.Encnter_DtTm
Data_Morte	Data morte	Followup.Expired_DtTm
Causa_Morte	Causa della morte	Followup.Cause_Death
Local	Indicatore Local	Followup.Local_Fail
Region	Indicatore Region	Followup.Region_Fail
Distant	Indicatore Distant	Followup.Distant_Fail
Tossicita1	Tipo di tossicità	Followup.Area1
Gravita1	Gravità	Followup.Severity1
Tossicita2	Tipo di tossicità	Followup.Area2
Gravita2	Gravità	Followup.Severity2
Tossicita3	Tipo di tossicità	Followup.Area3
Gravita3	Gravità	Followup.Severity3
tx_Intento	Tipo terapia	Prompt2.Description
tx_Tecnica	Tecnica terapeutica utilizzata	Prompt.Description

L'elenco delle tabelle utilizzate per la creazione della vista:

- dbo.Admin: tabella con parte dei dati anagrafici dei pazienti
- dbo.Patient: tabella con parte dei dati anagrafici dei pazienti
- dbo.Medical: tabella con i dati delle diagnosi
- dbo.Topog: tabella con la descrizione associata al codice malattia
- dbo.Morphol: tabella con la descrizione associata al codice morfologia
- dbo.TNMStage: tabella contenente i valori T,N,M e Stadio per diagnosi
- dbo.Followup: tabella con tutti i dati relativi al follow-up
- dbo.Prompt: tabella con dati relativi alla prescrizione di terapia
- dbo.PatCPlan: tabella con i dati in dettaglio per terapia
- dbo.Site: tabella con la codifica ICDO-9 per posizione della malattia

e i criteri con cui sono stati aggregati i dati presenti nelle tabelle:

```

FROM
dbo.Admin
INNER JOIN
dbo.Medical ON dbo.Admin.Pat_ID1 = dbo.Medical.PAT_ID1
AND dbo.Medical.Diagnosis_Class = 1
    
```

```

INNER JOIN
dbo.Patient ON dbo.Admin.Pat_ID1 = dbo.Patient.Pat_ID1
LEFT OUTER JOIN
dbo.Topog ON dbo.Medical.Topography = dbo.Topog.Diag_Code
LEFT OUTER JOIN
dbo.Morphol ON dbo.Medical.Histology = dbo.Morphol.ICDo_Code
LEFT OUTER JOIN
dbo.TNMStage
ON dbo.Medical.MED_ID = dbo.TNMStage.MED_ID
    AND dbo.TNMStage.StageType = '0'
LEFT OUTER JOIN
dbo.TNMStage AS TNMStage2 ON dbo.Medical.MED_ID = TNMStage2.MED_ID
    AND TNMStage2.StageType = '1'
LEFT OUTER JOIN
dbo.Followup ON dbo.Patient.Pat_ID1 = dbo.Followup.Pat_ID1
LEFT OUTER JOIN
dbo.PatCPlan ON dbo.Admin.Pat_ID1 = dbo.PatCPlan.Pat_ID1
LEFT OUTER JOIN
dbo.Site ON dbo.PatCPlan.PCP_ID = dbo.Site.PCP_ID
LEFT OUTER JOIN
dbo.Prompt ON dbo.Site.Technique = dbo.Prompt.Text
LEFT OUTER JOIN
dbo.Prompt AS prompt2 ON dbo.PatCPlan.Tx_Intent = prompt2.Text

```

Dalla sequenza di JOIN utilizzata si può notare come le tabelle "Prompt" e "TNMStage" siano state collegate due volte ciascuna per poter mettere su un unico record tutti i valori utili, inoltre come anticipato si nota che la tabella "Followup" viene collegata mediante il campo identificativo del paziente e non della diagnosi, andando così a generare dati non coerenti con le richieste.

2.2 Progettazione delle applicazioni e dei database

Considerando i software che formano il portale, la progettazione del database riguarda principalmente se non esclusivamente la gestione degli utenti che hanno accesso ai

programmi IRST tra cui il programma per la visualizzazione di una serie di indicatori e analisi statistiche per la radioterapia.

All'interno del portale di radioterapia non sono infatti, allo stato attuale, ancora previste funzionalità avanzate che richiedono il salvataggio di dati complessi gestiti da un database.

Al contrario tutti i dati necessari vengono importati da database esterni e elaborati al momento della richiesta dell'utente.

2.2.1 Gestioni permessi utenti

Il software di gestione degli utenti deve avere le seguenti caratteristiche:

- deve porsi in maniera trasparente per il programmatore dei singoli applicativi
- deve quindi poter gestire l'autenticazione a più programmi
- alcuni dei programmi potrebbero non necessitare di autenticazione
- deve poter essere integrato ad un server di autenticazione di tipo Active Directory
- deve poter gestire utenti al di fuori del dominio Active Directory
- nel caso non si utilizzi il dominio Active Directory deve gestire una complessità della password adeguata alle norme vigenti in materia con scadenza ogni 90 giorni
- deve permettere all'utente di recuperare le proprie credenziali in maniera autonoma
- deve permettere al programmatore di gestire diverse tipologie di utente all'interno del programma
- deve poter gestire le abilitazioni ai singoli programmi non solo per utente ma anche per gruppi formati da diversi utenti.

Per poter fare questo è necessario progettare un database in grado di soddisfare tutte le caratteristiche richieste oltre che sviluppare una libreria di funzioni da includere in tutte le pagine del server web.

Di seguito vengono proposte le tabelle create per la gestione degli utenti:

- UTENTI
id_utente, username, cognome, nome, codice fiscale, email,, email_alternativa
telefono, stato

- GRUPPI_UTENTI
id_gruppo, id_utente
- STORICO_PASSWORD
id_password, id_utente, password, data_inizio
- ABILITAZIONI
id_abilitazione, id_utente, id_applicazione, variabile, valore, descrizione
- APPLICAZIONI
id_applicazione, nome, descrizione, email_referente, persorso, immagine, tipo_accesso

Nella tabella "utenti" troviamo il campo identificativo *id_utente*, lo username utile nel caso l'utente non sia di dominio, il nome, il cognome, il codice fiscale, utile a identificare l'utente per la sincronizzazione dei dati con il dominio, la email principale (solitamente email dell'irst), una mail alternativa per il recupero della password, il numero di telefono e lo stato, lo stato dell'utente puo' assumere diversi valori: 10 se l'utente è attivo e sfrutta l'autenticazione da database, 20 se sfrutta l'autenticazione da dominio, 80 se l'utente corrisponde ad un gruppo e non ha quindi ne dati anagrafici ne password associata, 90 se l'utente è disabilitato.

"Gruppi_Utenti" è una tabella utile ad associare per ogni utente di tipo gruppo (stato 80) una serie di utenti o di gruppi, in questo modo è possibile creare anche gruppi di gruppi di utenti.

La tabella "Storico_Password" tiene traccia di tutte le password utilizzate dagli utenti che sfruttano l'autenticazione da database, in questo modo è possibile anche andare a gestire la possibilità di negare l'uso ripetuto della stessa password da parte di un utente per un certo numero di volte, il campo *id_password* è la chiave della tabella, *id_utente* è il campo utile per associare le password all'utente proprietario, *data_inizio* è un campo data dove viene inserita la data in cui la password è stata creata in modo da gestire la scadenza secondo un parametro non imposta da database.

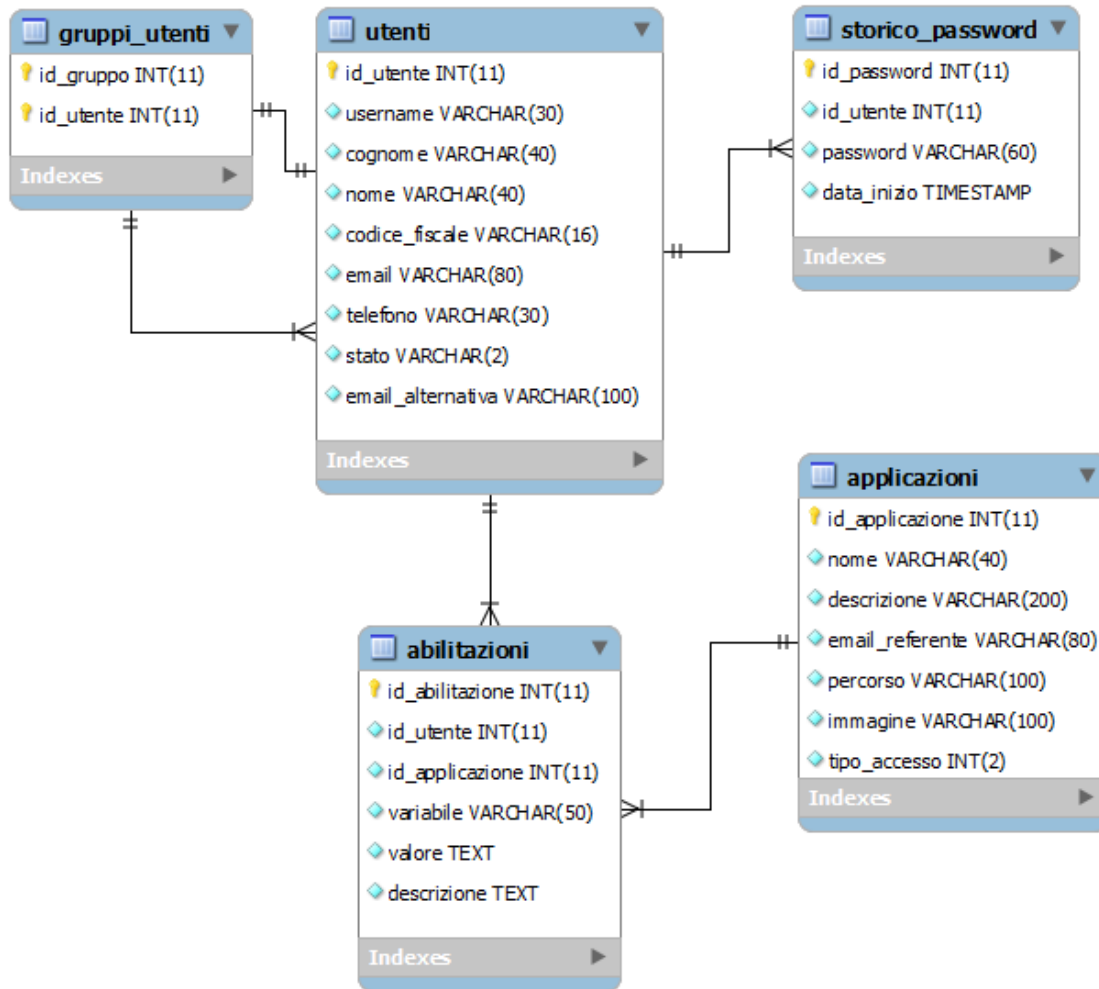


Figura 2.10 - Schema del database IRST per l'autenticazione degli utenti

La tabella "Abilitazioni" consente di mettere in relazione gli utenti o gruppi con le applicazioni mediante i campi `id_utente` e `id_applicazione` contenuti al suo interno, a questi affianca due campi utili alla definizione dei permessi, `variabile` e `valore` che il programmatore può sfruttare come preferisce all'interno della propria applicazione, è inoltre presente la chiave `id_abilitazione`.

Nella tabella "Applicazioni" troviamo l'anagrafica delle applicazioni registrate al programma, per ciascuna di esse è possibile salvare l'`id_applicazione`, il titolo dell'applicazione nel campo `nome`, una breve descrizione nel campo `descrizione`, l'indirizzo email del referente dell'applicazione, il campo `percorso` è utile al set di funzioni per identificare l'applicazione a cui si sta accedendo anche senza che il programmatore lo espliciti, un'icona nel campo `immagine` e il campo `tipo_accesso` che indica se l'applicazione necessita (valore 10) o meno (valore 0) della gestione utenti.

Ovviamente una qualsiasi applicazione non registrata su questa tabella verrà bloccata all'avvio restituendo un errore visivo.

2.3 Progettazione Interfaccia Utente

Nella progettazione dell'interfaccia per l'utente si è cercato di mantenere un aspetto il più possibile vicino ad applicazioni web molto diffuse, sebbene le funzionalità e le informazioni proposte dall'applicazione non siano esattamente quelle che un utente medio è abituato a utilizzare nel quotidiano.

L'obiettivo che ci si è prefissati è stato quello di rendere le schermate comprensibili e di facile utilizzo anche per una persona che non abbia idea del tipo applicazione che sta per andare a utilizzare, per ottenere questo risultato ci si è affidati alle tecnologie e alle librerie grafiche per web application più attuali andando a ricercare, a personalizzare o a creare da zero gli elementi grafici più adatti allo scopo.

L'informazione viene presentata all'utente con un'interfaccia *minimale e di facile utilizzo*, in modo che la sua comprensione dei pochi elementi presentati sia sufficiente a fargli capire la destinazione d'uso degli stessi.

per quanto concerne l'esecuzione è necessario ridurre il numero di passaggi per trasformare un'intenzione in un'azione.

Come detto in precedenza i software sviluppati o in via di sviluppo sono due, di seguito sono riportati vari *mock-up* utilizzati per la composizione dell'interfaccia utente in fase di programmazione.

2.3.1 Progettazione autenticazione utente

Lo scopo dell'applicazione di autenticazione è quello di fornire un software unico e scollegato dagli altri per la gestione delle credenziali dei vari servizi e applicativi sviluppati internamente.

Per l'autenticazione degli utenti si è pensato di utilizzare una schermata il più pulita possibile (Figura 2.11).

Al centro di una pagina color pastello, all'interno di una cornice con sfondo bianco, vengono posizionati i vari elementi: nella parte alta trova spazio il logo dell'ospedale, grande abbastanza per far capire immediatamente all'utilizzatore quali siano le credenziali da utilizzare e a quale portale sta tentando di accedere.

Posizionati subito sotto l'immagine i campi della form: nome utente, password e il bottone per inviare la richiesta di login.

Al di sotto del pulsante viene inserito un link per accedere alla pagina di ripristino della password nel caso l'utente se la sia dimenticata.

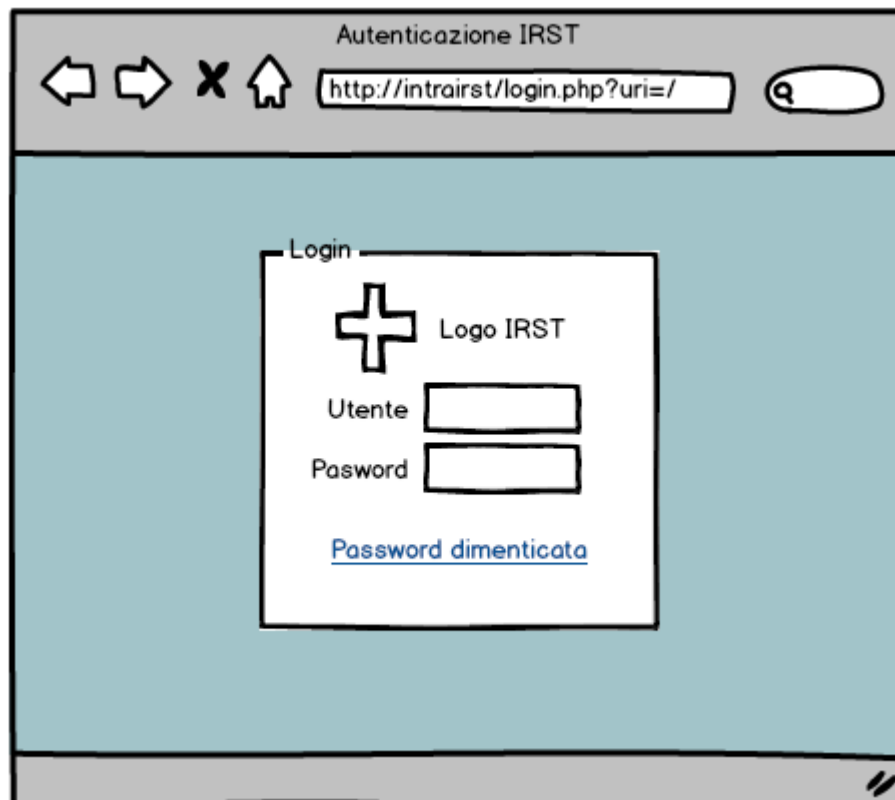


Figura 2.11 - Mockup schermata di login utente

Una volta effettuato il login l'applicazione di autenticazione rimane in qualche modo visibile all'utente sotto forma di una barra menù di colore celeste nella parte alta dell'applicazione desiderata.

All'interno di questa barra, partendo da sinistra troviamo il nome e il cognome dell'utente che ha effettuato l'accesso, questo dettaglio permette all'utilizzatore del *browser* di verificare in maniera semplice e veloce se l'utente connesso è il suo o quello di un altro.

Cliccando su questa prima parte comparirà una piccola finestrella in sovra impressione rispetto alla pagina visualizzata con indicati i propri dati e recapiti e una serie di *link* a funzionalità relative al servizio di autenticazione come ad esempio la possibilità di

modificare i dati relativi al proprio account, la possibilità di richiedere l'abilitazione a un programma IRST e il pulsante per effettuare il *logout*.

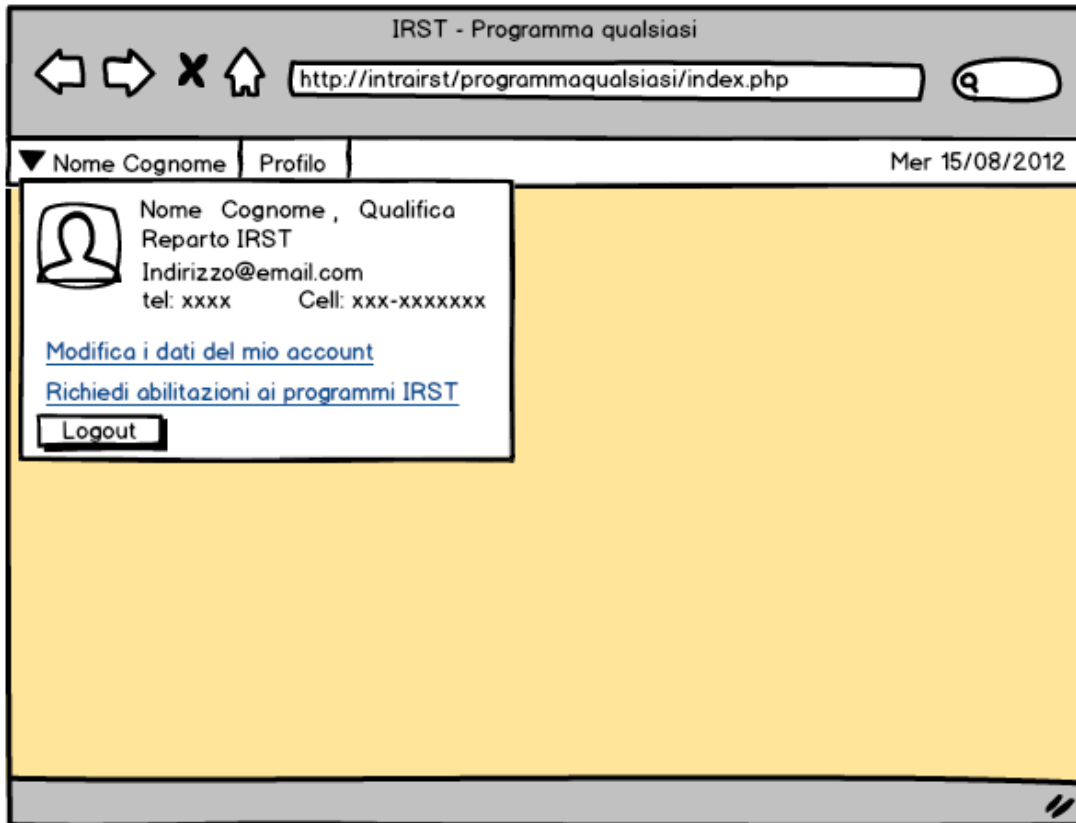


Figura 2.12 - Barra utente presente in ogni applicazione

Proseguendo verso destro troviamo prima il pulsante Profilo, questo pulsante è contestuale al programma che si sta visualizzando, se il programma lo prevede l'utente potrebbe avere contemporaneamente, permessi diversi sul software, cliccando su questo pulsante l'utente può variare il proprio status all'interno del programma inibendo l'utilizzo di certe parti del programma se il profilo selezionato possiede un livello di permessi più basso o viceversa abilitandone nel caso il profilo abbia permessi più elevati.

Nel caso in cui l'utente che ha fatto l'accesso sul software di autenticazione fosse un amministratore del software cliccando su questo pulsante verrà data la possibilità di caricare il profilo di un qualsiasi utente abilitato al programma allo scopo di fornire supporto allo stesso.

E nella parte più a destra troviamo invece la data odierna.

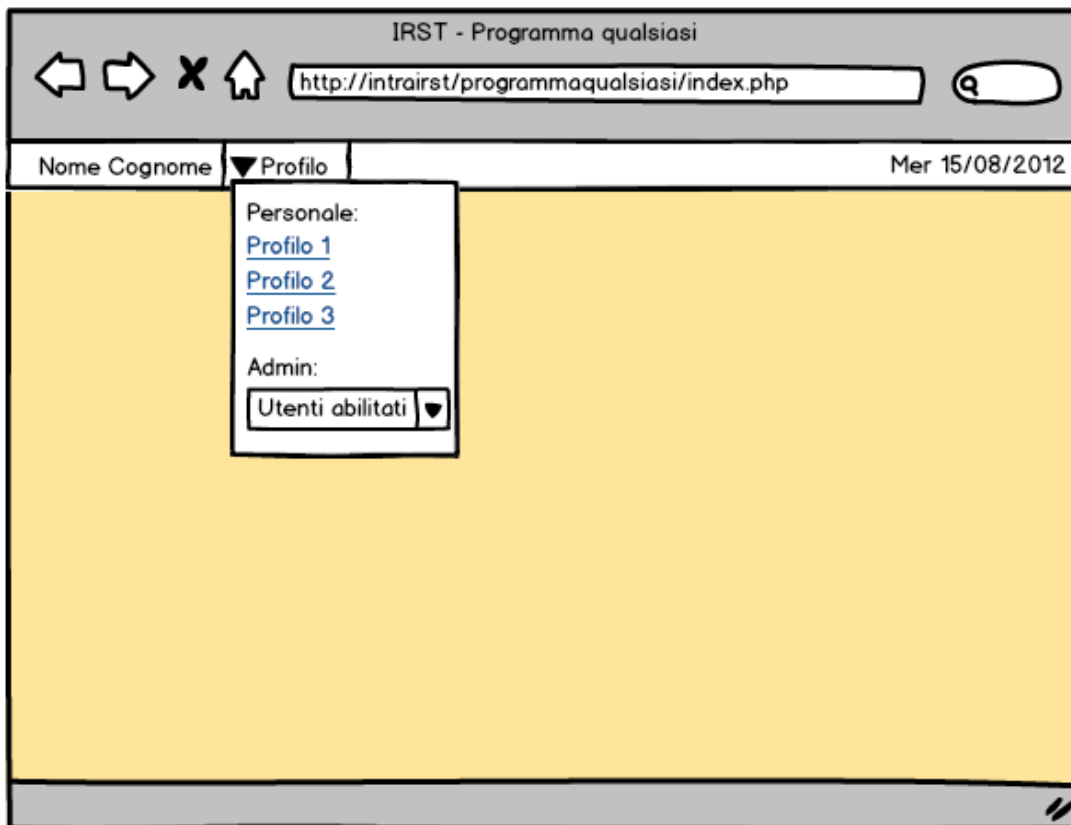


Figura 2.13 - Menu per la funzionalità di cambio profilo

2.3.2 Progettazione portale radioterapia

Essendo necessario mostrare un quantitativo di informazioni importante e abbastanza eterogeneo si è deciso di dividere il portale di radioterapia in tre pagine distinte.

Nella parte alta a sinistra è stato posizionato un menù orizzontale mentre a destra troviamo il nome identificativo del programma "Radioterapia IRST".

Cliccando su uno dei pulsanti nel menù di sinistra si accede alla sezione relativa ognuna delle quali presenta il proprio *layout* dedicato alle funzioni integrate al proprio interno.

2.3.3 Progettazione statistiche amministrative

Entrando nel programma di radioterapia si ha accesso diretto al pannello Monitor, in questa sezione troviamo sulla sinistra un menù con all'interno un piccolo form formato da due campi data (*date picker*) che ci permette di selezionare un lasso di tempo da analizzare.

Sulla destra invece troviamo un contenitore a cartelle, è stato scelto di utilizzare contenitori a cartelle per non espandere troppo la pagina in verticale in modo che l'utilizzatore possa avere tutte le informazioni relative alla estrazione desiderata all'interno di del video senza bisogno di scorrere la pagina.

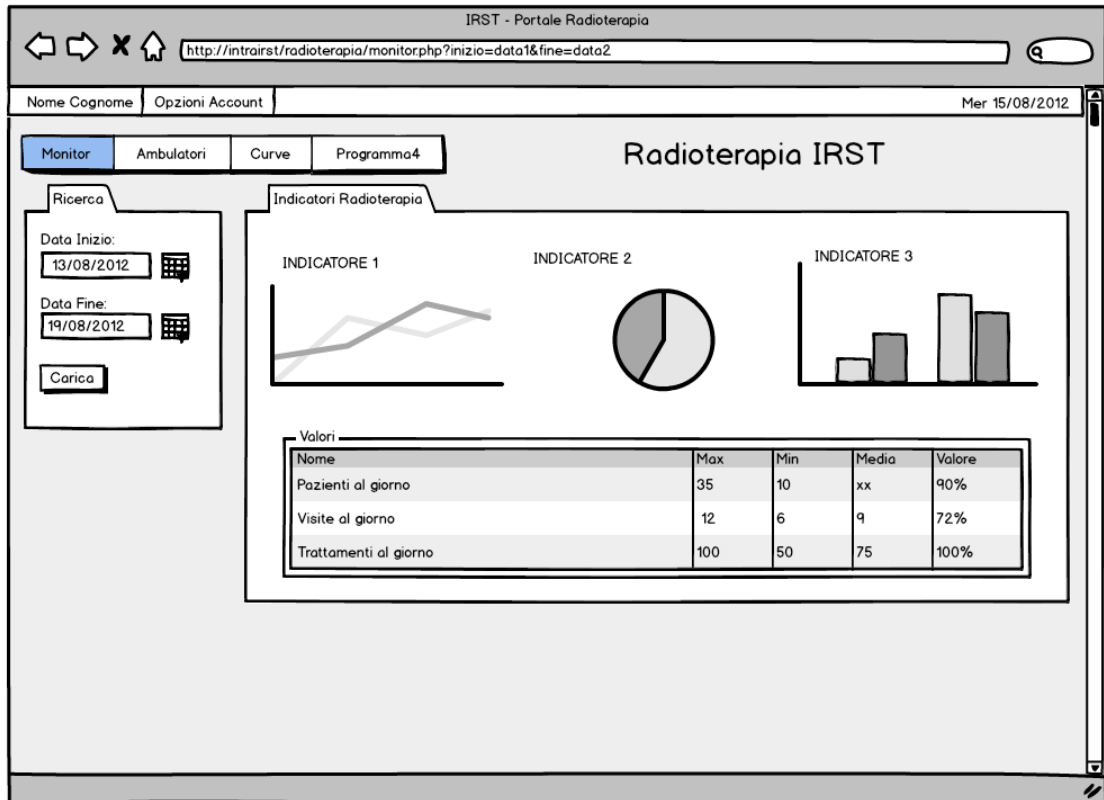


Figura 2.14 - Statistiche amministrative - Sezione Monitor

Allo stato attuale, non essendo ancora stato deciso in maniera definitiva quali saranno gli indicatori, è stata prevista una sola cartella all'interno della quale possono trovar spazio tre grafici di tipo diverso e una tabella riassuntiva.

2.3.4 Progettazione statistiche di reparto (ambulatori e macchinari)

Cliccando sul pulsante Ambulatori, è possibile entrare nel dettaglio dell'occupazione di ciascun ambulatorio o diagnostica, sulla parte sinistra troviamo sempre una piccola *form* con all'interno la possibilità di scegliere un lasso di tempo tra due date.

Sulla parte destra all'interno di un contenitore a cartelle troviamo i grafici e i relativi dati in tabella.

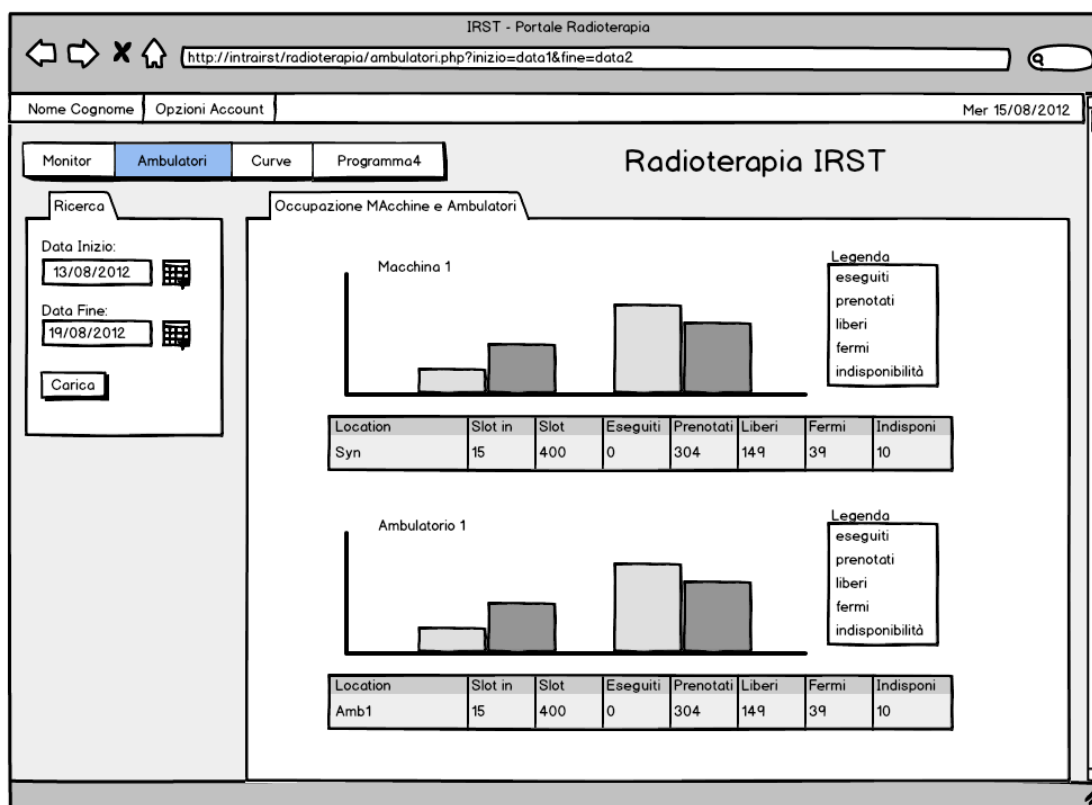


Figura 2.15 - Statistiche Ambulatori e Diagnostiche

Per la visualizzazione dei dettagli richiesti, si è scelto di utilizzare dei grafici a colonne, ogni colonna deve rappresentare un giorno del periodo selezionato, la colonna di ciascun giorno viene divisa in diverse porzioni, una per ogni tipologia di occupazione prevista, ad ogni tipologia corrisponde un colore diverso e all'interno della porzione viene indicato anche il numero corrispondente.

Sull'asse delle ordinate viene indicato con una linea l'occupazione massima prevista per ogni ambulatorio, in questo modo è possibile capire quando il carico di lavoro supera la programmazione prevista o viceversa quando non lo supera.

Sulla parte destra del grafico viene elencata in una legenda la corrispondenza tra il colore e il tipo di occupazione registrato mentre nella parte bassa è stata inserita una tabella riassuntiva utile per farsi un'idea complessiva in riferimento alle date selezionate.

2.3.5 Progettazione Curve di Sopravvivenza

Per la sezione relativa alle analisi delle curve di sopravvivenza, raggiungibile come le altre andando a cliccare nel menù alto sul pulsante relativo, si è deciso di mantenere una disposizione simile alle altre schermate.

Per questo motivo si è utilizzato per il menù di sinistra un contenitore a cartelle diviso in due parti, una prima parte dove compariranno tutti i campi utili a filtrare i dati dal database di reparto e una seconda parte dove sarà possibile raggruppare i dati estratti secondo certi criteri associando ad ogni raggruppamento un etichetta e un colore e così ridisegnare il grafico.

Secondo le richieste del medico il tipo di estrazione presenta due fasi: una nella quale viene creato il grafico partendo dai dati estratti secondo parametri presenti nel software di reparto.

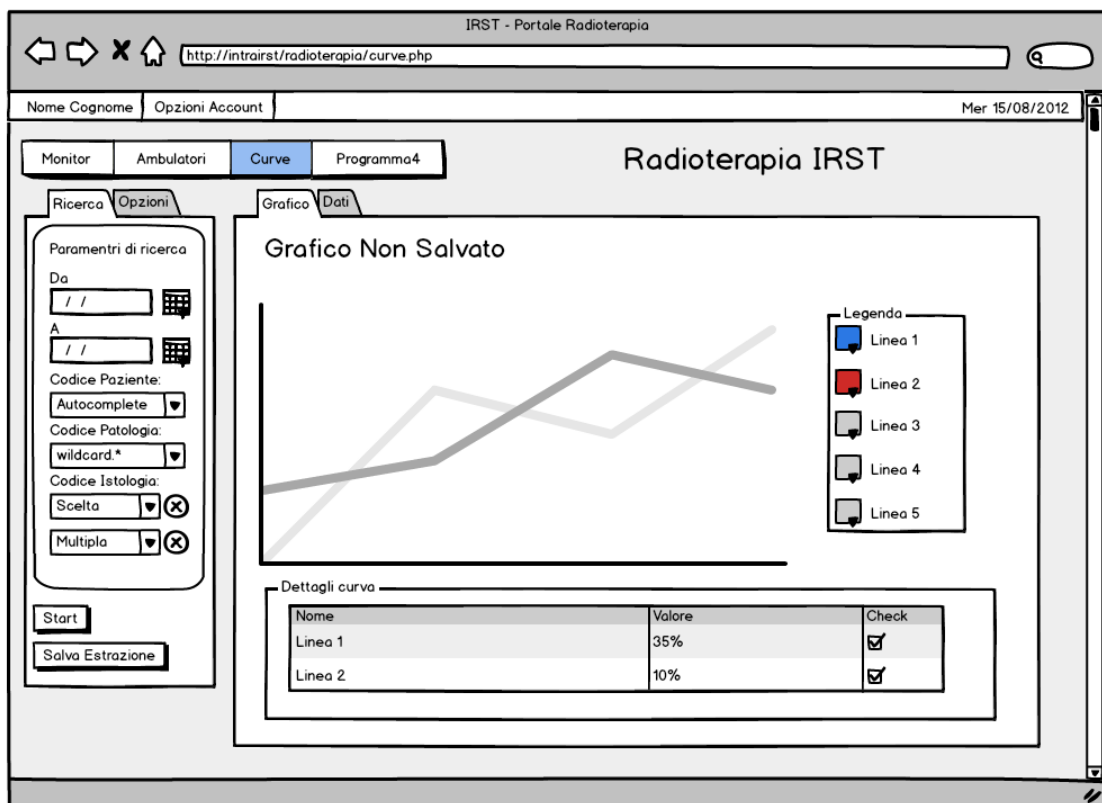


Figura 2.16 - Filtro per Curve di sopravvivenza

In questa fase come si può vedere dalla *figura 2.16* è presente sulla sinistra un menù che consente di selezionare, un periodo di tempo tramite l'utilizzo di un calendario per la selezione della data, e una serie di dati: codice paziente codice patologia e codice

istologia, per ciascuno dei quali è stata implementata la funzionalità di autocompletamento andando a ricercare i dati direttamente dal database di reparto oltre che a una serie di semplificazioni come l'utilizzo della *wildcard* asterisco e la possibilità di scelta multipla tramite l'aggiunta dinamica di campi testo dalle stesse caratteristiche. Nella seconda fase i dati estratti vengono rielaborati in modo da visualizzare raggruppamenti non noti a priori, con la possibilità di applicare al grafico una serie di accorgimenti estetici, come ad esempio il colore della curva.

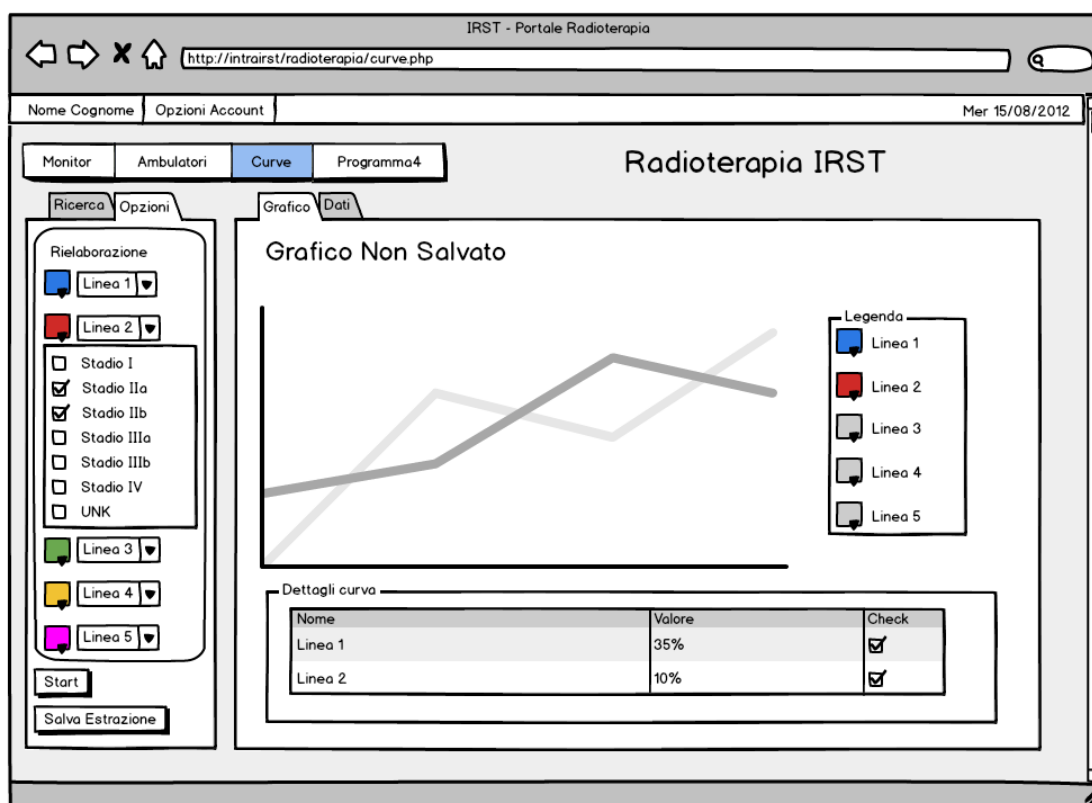


Figura 2.17 - Elaborazione estrazione Curve di sopravvivenza

Sulla parte sinistra della *figura 2.17* si può notare il menu per la rielaborazione del grafico.

In questo menu viene data la possibilità di raggruppare i dati estratti secondo uno specifico parametro, lo stadio della malattia.

Il menu fornisce la possibilità di creare più raggruppamenti, parametrizzabili tramite la selezione di *checkbox* per ogni raggruppamento, a cui seguirà poi la rappresentazione di una curva specifica sul grafico, è possibile applicare una etichetta testuale e un colore tramite la selezione da un comodo *color picker*.

Anche nel contenitore di destra sono presenti due cartelle, nella prima verranno inseriti il grafico e la legenda: il grafico dovrà essere come dalla teoria enunciata per le curve di sopravvivenza un grafico a scalino con i giorni sull'asse delle ascisse e un valore percentuale su quello delle ordinate.

Una funzionalità richiesta è quella di poter andare a "interrogare" il grafico in maniera interattiva per sapere all'istante il valore sull'asse delle Y corrispondente all'asse delle X. Sotto al grafico verranno poi tabellati i vari indicatori delle curve come la mediana di ciascuna curva e il logrank utile a comparare l'andamento di due curve di sopravvivenza.

Nella seconda cartella di destra vengono inseriti in una tabella tutti i dati utili a generare le curve secondo l'algoritmo Kaplan - Meier e associati a questi una serie di dati clinici e anagrafici utili sia ai medici che ai biostatistici.

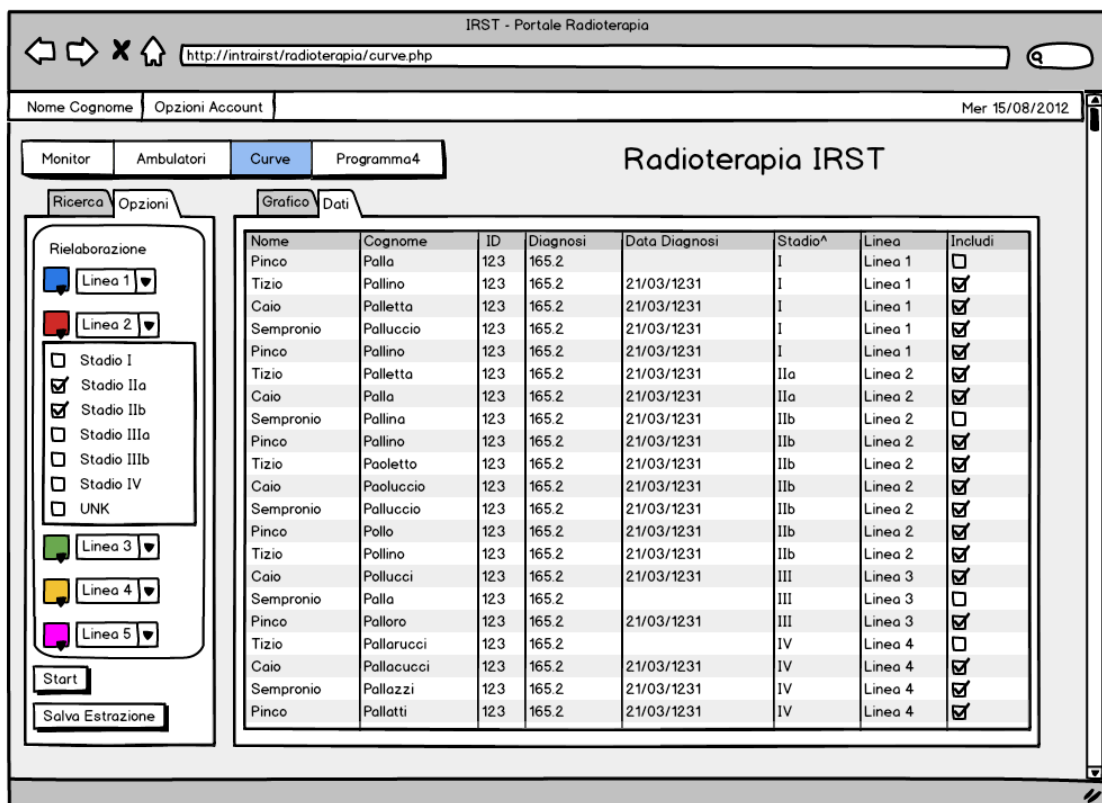


Figura 2.18 - Visualizzazione dati estratti Curve di sopravvivenza

Per i medici questa tabella fornisce un'indicazione precisa dei casi che vengono considerati all'interno del grafico e della singola curva rappresentata e offre la

possibilità di escludere singoli pazienti dalla curva a loro discrezione semplicemente cliccando sull'apposita *checkbox* posta in fondo alla riga.

Tramite la possibilità di export in formato CSV o Excel, questa tabella offre ai biostatistici la possibilità di verificare con software dedicati se la curva e i valori calcolati siano corretti andando così ad effettuare un controllo di qualità sul software che ha generato le curve.

2.4 Tecnologie utilizzate per lo sviluppo

In questo paragrafo si analizzeranno le varie tecnologie utilizzate per lo sviluppo del software facendo una breve introduzione per ciascuna di esse e andando poi ad esaminare le caratteristiche principali e le peculiarità utilizzate all'interno del progetto per ottenere il risultato richiesto.

2.4.1 PHP

Acronimo di (*Hypertext Preprocessor*), PHP è attualmente il linguaggio di programmazione per la creazione di siti web più diffuso su internet [WTS13].

Lo scopo del linguaggio è quello di consentire agli sviluppatori web di realizzare in modo veloce pagine dinamiche.

Per pagine dinamiche si intendono pagine il cui contenuto viene, generato nel momento in cui le stesse vengono richieste al web server.

Le peculiarità del linguaggio sono:

- è un linguaggio interpretato:

I programmi scritti in questo linguaggio sono denominati brevemente *script* e vengono eseguiti tramite un apposito software, l'interprete PHP.

Quest' ultimo si occupa di leggere il codice contenuto all'interno dello *script* e, interpretandone le istruzioni, esegue le operazioni corrispondenti come ad esempio la lettura di un file o un calcolo aritmetico.

Dunque il PHP è quello che tecnicamente si definisce un *linguaggio interpretato* e in questo esso si differenzia da altri linguaggi di programmazione, come ad esempio C++ e Java, il cui codice sorgente, per poter essere eseguito, deve

prima essere compilato (tradotto cioè in codice macchina o nel caso di java in codice per la macchina virtuale).

- è HTML embedded:

Dalla documentazione ufficiale il PHP viene definito come un *linguaggio script lato server immerso nel codice HTML* [PHP13].

Questa caratteristica si riferisce al fatto che il codice PHP è inserito all'interno del codice HTML e viene riconosciuto dall'interprete mediante l'utilizzo di marcatori dedicati.

Il web server riconosce le pagine PHP, distinguendole da quelle "statiche", sulla base dell'estensione, che non sarà .htm o .html ma piuttosto .php, .php3 o simile; quando il server riconosce una estensione associata a PHP passa il testimone all'interprete, lasciando che sia quest'ultimo ad occuparsene.

- è server-side:

Il PHP opera lato server, ciò vuol dire che tutta l'elaborazione di uno script avviene sul server, prima che questi spedisca la pagina al *browser*.

Di conseguenza, chi accede ad una pagina PHP non ha la possibilità di leggere le istruzioni in essa contenute ma essendo state processate, ciò che il client vedrà sarà solamente il risultato dell'elaborazione.

La necessità di generare automaticamente il contenuto di pagine web si presenta in molte situazioni: si pensi a un qualsiasi sito internet che fornisce in tempo reale gli indicatori dell'andamento del mercato azionario, o i risultati della giornata sportiva, o la situazione del traffico automobilistico, etc.

Nel caso specifico la necessità di generare automaticamente delle pagine web è legata alla possibilità di interrogare i database contenenti i dati amministrativi e clinici relativi al reparto di radioterapia IRST.

In questa situazione occorre quindi avere a disposizione uno strumento che ci consenta appunto la comunicazione con un database e soprattutto la possibilità di manipolare le informazioni e presentarle in vari formati partendo ovviamente dal HTML fino al XML o al JSON.

PHP viene distribuito sia sotto forma di codice sorgente, da compilare, sia in formato binario precompilato; le versioni binarie vengono comunque rese disponibili sia per la piattaforma Microsoft Windows, che per quella Linux in formato rpm.

L'installazione dell'interprete PHP può avvenire secondo due approcci alternativi, che differiscono tra loro per la modalità di interazione con il server web.

Il primo approccio consiste nell'installazione di PHP sotto forma di applicazione CGI; il secondo nell'installazione sotto forma di modulo del server web.

Nell'installazione di tipo CGI, l'interprete PHP verrà invocato dal server web come qualsiasi altra applicazione in grado di comunicare con esso attraverso la *Common Gateway Interface*, l'interprete, sarà dunque considerato come un programma *esterno* al server web.

In questa situazione l'esecuzione di uno script PHP consiste in una invocazione dell'interprete al quale viene passato il percorso del file da eseguire tramite la variabile d'ambiente `PATH_INFO`.

Questo tipo di installazione è il più semplice ma comporta alcuni svantaggi, sia in termini di prestazioni che in termini di funzionalità.

Le prestazioni vengono influenzate negativamente dalla necessità di creare un nuovo processo ogni volta che viene invocato uno script PHP, con un considerevole aggravio del carico di lavoro della macchina server. Inoltre, alcune funzionalità del linguaggio, in particolare quelle che richiedono una interazione più profonda con il server web, sono disponibili solo quando PHP viene utilizzato come modulo.

L'interprete PHP può essere integrato nel server web sotto forma di modulo statico o dinamico: nel primo caso (modulo *statico*) sarà necessario effettuare la ricompilazione del server web ed il risultato ottenuto sarà un nuovo eseguibile, completo di supporto PHP; nel secondo caso (modulo *dinamico*) l'interprete PHP verrà compilato come libreria a caricamento dinamico: *DSO (Dynamic Shared Object)* su piattaforme Unix e *DLL (Dynamic Link Library)* su piattaforme Windows [PHP13].

Notiamo che l'esecuzione di uno script PHP non richiede l'invocazione di un programma esterno (e quindi la creazione di un nuovo processo, come avveniva nell'ipotesi di installazione CGI) in quanto viene effettuata *all'interno* del server web.

L'installazione di PHP sotto forma di modulo del server web è quella che fornisce le prestazioni più elevate, peraltro senza rinunce in termini di funzionalità disponibili; tale approccio è stato quindi preferito nel contesto dell'applicativo reale.

Oltre a queste caratteristiche il PHP offre una varietà di funzionalità e vantaggi e di seguito ne sono elencati i principali:

- l'interprete PHP (o processore di ipertesto PHP) può essere integrato facilmente nel web server Apache come modulo ed è piuttosto leggero.
- è disponibile gratuitamente (i dettagli della licenza si trovano sul sito, oppure nel file LICENSE della distribuzione in sorgente);
- la portabilità: utilizzando piccoli accorgimenti PHP è in grado di funzionare su diverse piattaforme (Windows, Unix/Linux, Mac OS e altri) ed è in grado di appoggiarsi indifferentemente a diversi web server (Apache, IIS di Microsoft, OpenHTTPd, ecc...) grazie all'interfaccia CGI (Common Gateway Interface).
- è già fornito di tutte le librerie per accesso ai database più comuni e di tutte le funzioni per la gestione dei cookies, delle sessioni di navigazione, per l'utilizzo di protocolli di rete, per la generazione dinamica delle immagini e di tutto quello di cui abbiamo bisogno per programmare pagine web dinamiche anche complesse.
- facilita il debugging: fa uscire gli errori direttamente sulle pagine web e/o nell'errorlog, indicando chiaramente il tipo di errore, il percorso completo dello script che ha generato l'errore e il numero di riga su cui si è fermato il parser.
- può essere utilizzato anche come linguaggio di scripting al di fuori del web, ad esempio per produrre script destinati ad essere chiamati dalla shell.

Predisporre una piattaforma completa installando su sistema operativo Linux il web server Apache, PHP e un motore di database come MySQL o PostgreSQL è a costo zero.

Nel caso specifico andandoci ad appoggiare ad una serie di sistemi già esistenti il server Apache e il PHP sono stati installati su una macchina fisica con sistema operativo Windows Server 2003r2 di Microsoft utilizzata all'interno dell'ospedale per fornire una serie di servizi basati sul web.

Il database per la gestione degli utenti risiede su un server MySQL su sistema operativo Linux virtualizzato, mentre l'interazione con i dati provenienti dal software amministrativo (LOG80) e da quello clinico di reparto (Mosaiq) sono diffusi tramite l'utilizzo di viste dedicate rispettivamente sui server Linux/MySQL per LOG80 e sul server Windows 2003r2/SQL Server per quanto riguarda il software Mosaiq.

2.4.2 AJAX

L'acronimo AJAX, significa *Asynchronous JavaScript And XML* (JavaScript asincrono ed XML), è stato enunciato per la prima volta da Jesse Garrett, nel 18 Febbraio 2005, come titolo di un post all'interno del suo blog [GAR05].

Non si tratta di una nuova tecnologia né di un'invenzione ma di un concetto utilizzato per sviluppare applicativi avanzati e particolari quali i servizi di email online più moderni fino ad arrivare ad applicazioni, come nel nostro caso, basate sulle tecnologie web.

Il concetto è in parte espresso nell'acronimo scelto, un utilizzo asincrono di Javascript che attraverso l'interfacciamento con XML, può permettere a un client di richiamare informazioni lato server in modo veloce e trasparente, incrementando le prospettive delle applicazioni web dinamiche, che erano precedentemente legate a tecnologie principalmente client side come Adobe Flash o alle più complete applet Java, le quali necessitavano però di un software addizionale rispetto al solo client e la loro diffusione era quindi condizionata dalle competenze tecniche del singolo utente oltre che dalla compatibilità con il client web.

Una prima intuizione in merito venne fornita agli sviluppatori web con gli *iframe*, che consentivano di emulare una certa interazione tra client e server andando a caricare contenuti esterni all'interno di una porzione della pagina web in maniera asincrona e cioè ottenendo un primitivo effetto di aggiornamento della pagina senza ricaricarla completamente.

Solo in un secondo momento, Microsoft sviluppò una tecnologia chiamata *Remote Scripting*, con lo scopo di creare una tecnica più elegante per richiamare contenuti differenti ed è in questo periodo, seppur con un altro nome, che AJAX venne utilizzato per la prima volta, andando poi evolvendosi in versioni più mature fino a diventare un oggetto vero e proprio, *XMLHttpRequest*.

Fu però grazie all'utilizzo da parte di Google che la maggior parte degli addetti ai lavori poté comprendere appieno le potenzialità offerte da questa tecnologia.

Applicazioni come *gmail* e *google maps* rivoluzionarono il panorama degli applicativi web e affondarono il loro successo proprio fatto che non necessitavano l'utilizzo di plugin come flash player o le java applet, il che facilitava la fruizione da parte del grande pubblico.

Altro fattore molto importante essendo AJAX basato su javascript, tecnologia principalmente utilizzata per la manipolazione di codice HTML lato client, forniva la possibilità di considerare il fattore dell'accessibilità fino a quel momento ignorato dalle altre tecnologie.

L'oggetto AJAX implementato allo stato attuale su tutti i browser principali, permette di effettuare la richiesta di una risorse tramite HTTP a un server web in modo indipendente dal browser stesso.

Nella richiesta è possibile inviare informazioni tramite l'uso di variabili come lo si farebbe con una form HTML utilizzando indifferentemente i metodi GET o POST.

La richiesta, come indicato dall'acronimo stesso, è di tipo asincrono, il che significa che non è necessario attendere la risposta del server per procedere ed effettuare altre operazioni, stravolgendo così, sotto diversi punti di vista, il flusso di dati tipico di una pagina web.

Quello che accade in assenza di richieste AJAX è infatti che l'utente interagisce con la pagina web tramite l'utilizzo di link o form che generano una richiesta al web server il quale elabora o più semplicemente ripropone la pagina richiesta e solo quando questa è visualizzata nel browser l'utente può passare eventualmente a interagire nuovamente con l'applicazione.

Adottando AJAX si perde questa linearità (richiesta e risposta) e mentre una richiesta viene elaborata dal server l'utente o l'applicazione stessa è libera di effettuare altre richieste simultanee al server per operazioni differenti.

Il tipo di risposta che l'oggetto si aspetta dopo una chiamata non è necessariamente di tipo XML ma, in contro tendenza con l'acronimo stesso, è di tipo testuale.

All'interno del progetto tuttavia le chiamate ajax sono interamente effettuate tramite l'oggetto omonimo \$.ajax() implementato con le librerie jQuery e utilizzano principalmente il formato JSON per la trasmissione dei dati sia in uscita che in ingresso.

2.4.3 JSON

JSON o *JavaScript Object Notation* [JSO13] è un formato per lo scambio di dati semplice da comprendere e da generare, ed è anche uno dei modi con cui è possibile definire gli oggetti nel linguaggio *Javascript*.

E' un formato di testo completamente indipendente dal linguaggio di programmazione, ma utilizza convenzioni conosciute dai programmatori di linguaggi della famiglia del C, come C, C++, C#, Java, JavaScript, Perl, Python, e molti altri.

Questa caratteristica fa di JSON un linguaggio ideale per lo scambio di dati.

Il formato si basa su due strutture:

- un insieme di coppie nome/valore. In diversi linguaggi è realizzato tramite un oggetto, un *record*, una *struct*, un dizionario, una tabella *hash*, un elenco di chiavi o un *array* associativo;
- un elenco ordinato di valori. Nella maggior parte dei linguaggi questo si realizza con un *array*, un vettore, un elenco o una sequenza.

Queste sono strutture di dati universali, virtualmente tutti i linguaggi di programmazione moderni li supportano in entrambe le forme ed è quindi sensato che un formato di dati che sia interscambiabile con i linguaggi di programmazione debba essere basato su queste strutture.

Un oggetto JSON è una serie non ordinata di nomi/valori. Un oggetto inizia con { (parentesi graffa sinistra) e finisce con } (parentesi graffa destra). Ogni nome è seguito da : (due punti) e la coppia di nome/valore sono separata da , (virgola).

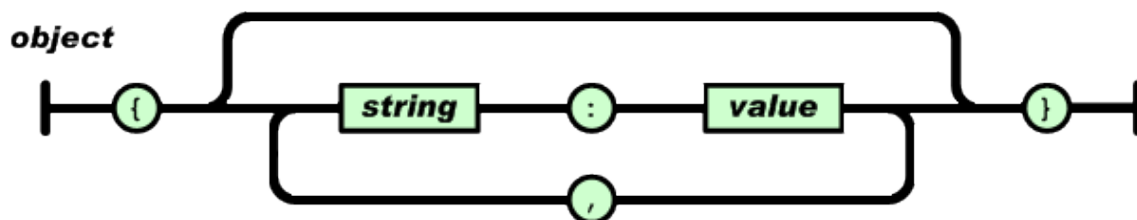


Figura 2.19 - Oggetto JSON

L'array JSON è una raccolta ordinata di valori, un *array* comincia con [(parentesi quadra sinistra) e finisce con] (parentesi quadra destra) mentre i valori al suo interno sono separati dal carattere ,(virgola).

Un valore può essere una stringa tra virgolette, un numero, un booleano, un oggetto o un *array*, queste strutture possono essere annidate.

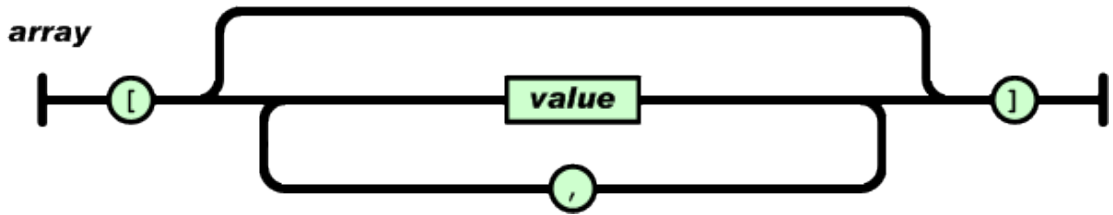


Figura 2.20 - Array JSON

Un valore può essere una stringa tra virgolette, o un numero, o vero o falso o nullo, o un oggetto o un array. Queste strutture possono essere annidate.

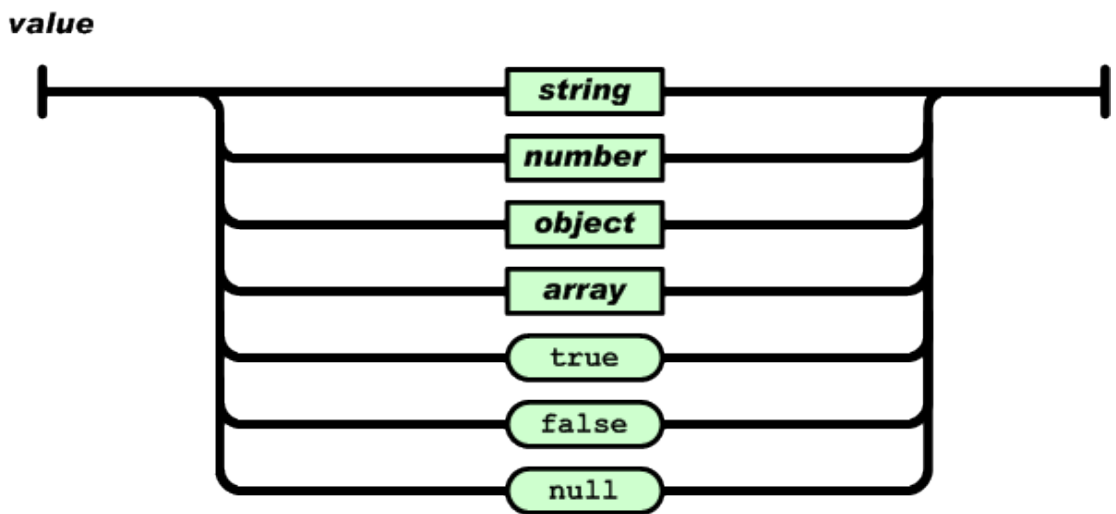


Figura 2.21 - Tipi di valore in JSON

Un numero è molto simile ad un numero C o Java, a parte il fatto che i formati ottali ed esadecimali non sono utilizzati.

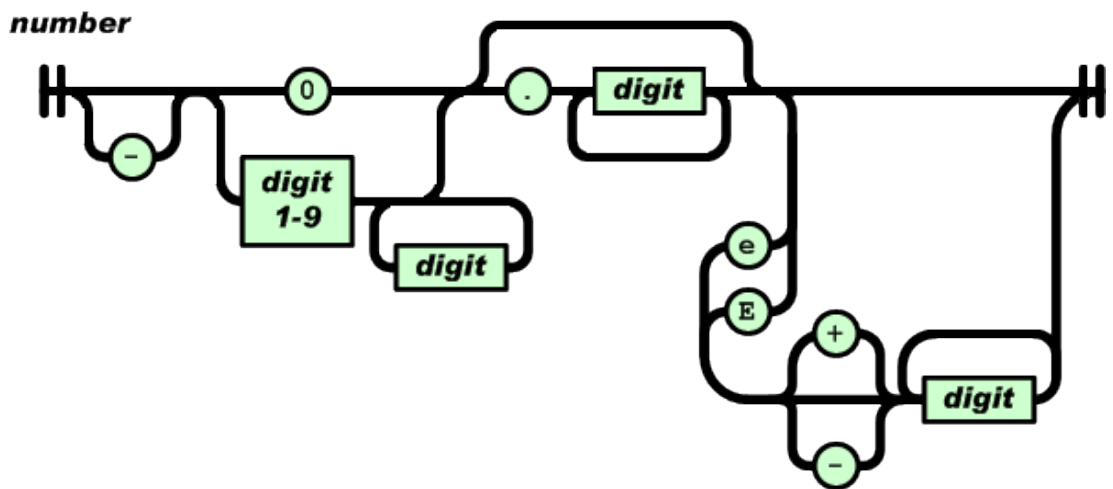


Figura 2.22 - Tipo numero JSON

Concludo mostrando un esempio di dati JSON:

```
{ "widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}
```

2.4.4 XML e XHTML

XML acronimo di *extensible markup language*, è un insieme di regole sintattiche per modellare la struttura di documenti e dati.

Questo insieme di regole, dette più propriamente specifiche, definiscono la modalità secondo cui è possibile strutturare e veicolare i propri dati.

E' definito meta-linguaggio di markup, e cioè un linguaggio che permette la definizione di altri linguaggi di markup.

Nel progetto è stato utilizzato XML principalmente per la comunicazione dei dati da graficare ed eventualmente salvare tra le varie pagine scritte in PHP.

Questo accorgimento è stato utilizzato perché è possibile che nel futuro il programma possa integrarsi con altri software esterni tramite l'utilizzo di web service, ad esempio per inviare informazioni relative alla primo spazio utile per prenotare una visita, o alla quantità di lavoro svolto in reparto, oltre che a poter ricevere dati da servizi esterni elaborati ad hoc per poi graficare i risultati secondo le modalità studiate nella radioterapia IRST.

Le stesse informazioni precedentemente presentate in formato JSON in XML assumono questo aspetto:

```

<widget>
  <debug>on</debug>
  <window title="Sample Konfabulator Widget">
    <name>main_window</name>
    <width>500</width>
    <height>500</height>
  </window>
  <image src="Images/Sun.png" name="sun1">
    <hOffset>250</hOffset>
    <vOffset>250</vOffset>
    <alignment>center</alignment>
  </image>
  <text data="Click Here" size="36" style="bold">
    <name>text1</name>
    <hOffset>250</hOffset>
    <vOffset>100</vOffset>
    <alignment>center</alignment>
    <onMouseUp>
      sun1.opacity = (sun1.opacity / 100) * 90;
    </onMouseUp>
  </text>
</widget>

```

Come già detto XML server per definire linguaggi di markup e XHTML è uno dei casi. XHTML come intuibile dal nome è la versione ridefinita secondo le specifiche XML di HTML, le differenze con html sono per lo più sintattiche, ad esempio: tutti i tag e i loro attributi devono essere espressi in minuscolo, è obbligatorio inserire i tag di chiusura e per i tag vuoti è necessario seguire la sintassi minimizzata (a capo in HTML è
 mentre in XHTML diventa
), i valori degli attributi devono essere inseriti tra apici singoli o doppi è necessario utilizzare l'attributo *id* al posto di name per identificare gli elementi del documento.

Queste piccole modifiche consentono ai browser ma soprattutto ad altri software come gli indicizzatori utilizzati dai motori di ricerca, i nuovi dispositivi connessi al web ma

soprattutto ai sintetizzatori vocali di elaborare più facilmente le pagine scritte in XHTML.

Altra cosa molto importante XHTML appoggiandosi a XML è riconosciuto come standard e questo garantisce una certa indipendenza da una specifica piattaforma hardware o software ed è per questo motivo che attorno ad XML ruota una serie di tecnologie e linguaggi come ad esempio CSS, XSL per la presentazione dei dati, DTD e XML-Schema grazie ai quali è possibile definire formalmente la "grammatica" di un nuovo linguaggio di markup fino a XQuery e XQL per l'estrazione di informazioni secondo determinati criteri da un documento XML.

2.4.3 JavaScript e JQuery

JavaScript è un linguaggio di programmazione orientato agli oggetti comunemente usato nei siti web, fu sviluppato da Netscape, azienda pioniera del web.

La sintassi è stata volontariamente resa simile a Java per facilitarne la diffusione, e dato il notevole successo ottenuto da questo linguaggio anche Microsoft sviluppò un linguaggio compatibile chiamato JScript.

Nonostante il linguaggio venne reso standard ufficialmente (ECMA 262), ogni browser ha sempre continuato ad avere una propria implementazione di JavaScript, spesso costruita per funzionare solo all'interno di esso.

La mancanza di uno standard di fatto porta a una notevole mole di lavoro per gli sviluppatori che devono testare il codice su tutti i browser per avere una compatibilità più estesa possibile, se oltre allo sviluppo di codice JavaScript c'è poi la necessità di sviluppare funzionalità AJAX il compito di mantenere la massima compatibilità diventa molto dispendioso.

Per aggirare questo ostacolo sono nati negli anni diversi *framework* basati su JavaScript con l'intento di fornire agli sviluppatori un set di strumenti pronti all'uso e che fossero uguali per tutti i *browser*.

jQuery [JQF13a] è un *framework JavaScript* che semplifica la definizione di comportamenti dinamici *client side* da parte dell'utente e l'utilizzo di AJAX.

Il *framework* fornisce metodi e funzioni per gestire al meglio aspetti grafici e strutturali come posizione di elementi, effetto di *click* su immagini, manipolazione del *DOM* e quant'altro, mantenendo la compatibilità tra *browser* diversi e standardizzando gli oggetti messi a disposizione dall'interprete JavaScript del *browser*.

E' inoltre possibile utilizzare dei *plugin* per estenderne le funzionalità in ogni aspetto, come il *plugin* ufficiale *jQuery User Interface* [JQF13b] che fornisce la possibilità di utilizzare componenti grafiche (quali navigazione a *tab*, *scroll-bar*, calendari, *drag and drop*, *input box* con funzionalità di *autocomplete* ecc.) per avvicinare l'esperienza di utilizzo di un'applicazione *web* a quella di un'applicazione *desktop*.

L'interfaccia del sito sviluppato in questo progetto sfrutta principalmente jQueryUI e tutti i suoi componenti, andando ad affiancare una serie di *plugin* personalizzati per lo scopo e controlli sviluppati ad hoc.

Tra i *plugin* jQuery, si è fatto largo uso di *dataTable* per la rappresentazione di tabelle interattive, questo *plugin* nella sua implementazione standard permette di rappresentare dati con un aspetto gradevole e di interagire con essi ad esempio ordinando la tabella su una o più colonne, molto importante è anche la funziona di ricerca e la possibilità tramite le API fornite dal *plugin* stesso di interagire con i dati rappresentati tramite azioni quali il click e il doppio click [SPR13].

Un altro *plugin* utilizzato è *Flot*, scritto interamente in javascript con l'intento di fornire uno strumento per la rappresentazione di grafici per jQuery, di semplice utilizzo, interattivo e dall'aspetto accattivante [LAU13].

Questo *plugin* è stato scelto per la rappresentazione delle curve di sopravvivenza, grazie al supporto nativo ai grafici a scalino e a una serie di funzionalità interattive implementate nel programma come ad esempio la possibilità di nascondere singole serie di punti e l'interazione con gli assi utili a mostrare in corrispondenza di un punto sull'asse delle ascisse il valore di tutte le curve rappresentate semplicemente tenendo traccia della posizione del mouse all'interno del grafico.

CAPITOLO 3.

IMPLEMENTAZIONE

In questo capitolo andiamo ad analizzare, modulo per modulo, il set di funzioni lato *server* che permettono la corretta gestione delle informazioni e le soluzioni adottate nella realizzazione dell'interfaccia grafica.

3.1 Modulo gestione Utenti

Questo modulo consiste principalmente di una libreria di funzioni php utilizzate per l'autenticazione e la gestione degli utenti e una serie di pagine web, come la pagina di registrazione, di inserimento delle credenziali, di recupero password e la barra applicazione mostrata nel *paragrafo 2.3.1*.

3.1.1 Set funzioni PHP

A disposizione del portale dedicato ma anche dei programmatori delle singole applicazioni web che risiedono e risiederanno sul web server IRST, sono state sviluppate una serie di funzioni utili per l'interrogazione del *database*, analizzato nel *paragrafo 2.2.1*, e per l'interazione con il server di dominio *Active Directory*, in modo da creare un'interfaccia trasparente per il programmatore che in questo modo, imparate le poche funzioni a disposizione, potrà concentrarsi sullo sviluppo del software senza curarsi della parte di autenticazione.

Questa libreria di funzioni viene inclusa automaticamente all'apertura di ogni pagina grazie all'opzione del *file* di configurazione di PHP:

```
auto_prepend_file = "C:\webapps\lib\irst\irst.inc.php"
```

All'interno del *file irst.inc.php* vengono inclusi pochi file, inizialmente viene effettuato un controllo per verificare che il programma aperto, necessiti di autenticazione o meno, questo è possibile farlo analizzando la variabile di *Apache* che mostra il *path* della pagina richiesta dal client.

Nel caso fosse necessario include poi la pagina dove è implementata la logica di funzionamento.

Questa pagina verificherà le credenziali dell'utente e ridirigerà lo stesso all'applicazione desiderata.

File `irst.inc.php`:

```
//se la pagina è caricata tramite web server
if (isset($_SERVER['HTTP_USER_AGENT'])){
    //include la libreria per l'utilizzo dei database
    require_once($_SERVER["DOCUMENT_ROOT"].'/lib/db/db.inc.php');
    //include una funzione per verificare i dati dell'applicazione
    require_once('include/getApplicazione.php');
    //carica i dati relativi all' applicazione
    $applicazione = getApplicazione("all");
// se l'app prevede un accesso diverso da 0 (nessuna autenticazione)
if ($applicazione['tipo_accesso']!='0'){
    //include la pagina dove è implementata la logica
    require_once ('irst.base.php');
}
}
```

All'interno della pagina `irst.base.php` vengono incluse tutte le altre funzioni utili alla gestione dei dati presenti nel database di gestione degli utenti e delle applicazioni.

Di seguito riporto tutte le funzioni incluse con una breve descrizione dell'utilità, i parametri richiesti in ingresso e quelli restituiti in uscita:

```
setNuovoProgramma ($nome, $descrizione, $email_referente, $percorso,
$immagine, tipo_accesso)
```

prende in ingresso tutti i parametri utili all'inserimento su database di un nuovo applicativo (nome, descrizione, email_referente, percorso, immagine, tipo accesso), in risposta presenta l'id associato all'applicazione in caso di successo, torna invece *false* in caso l'inserimento abbia violato qualche vincolo imposto dal database.

```
nuovoUtente ($cognome, $nome, $cf, $email='', $email_alternativa='',
$telefono='')
```

Serve ad inserire, e in questo caso anche ad aggiornare, i dati relativi ad un utente o a un gruppo, non imposta la password.

Al suo interno è definita una sotto funzione utile alla generazione dello username partendo dal nome e cognome dell'utente, questa funzione semplicemente sostituisce i caratteri speciali presenti nel nome con altri caratteri dell'alfabeto inglese, verifica che

non ci sia un utente con lo stesso username presente in archivio ed eventualmente, nel caso in cui i dati anagrafici non coincidessero aggiunge un indice numerico come ultimo carattere dello username.

La funzione nuovo utente restituisce tutti i dati dell'utente appena creato: id, nome cognome, codice fiscale, username o *false* nel caso di errore.

```
setAbilitazione ($id_utente, $id_Applicazione, $variabile, valore="",
descrizione="")
```

Inserisce o modifica, una nuova riga nella tabella "Abilitazioni" torna *true* o *false*.

```
deleteAbilitazione (id_utente, $id_Applicazione, $variabile='')
```

Elimina una o più righe di abilitazione prendendo in ingresso `id_utente` e `id_applicazione` come parametri obbligatori e opzionalmente il parametro chiamato `variabile`.

```
getUtenti($id_utente='', $username='', $cognome='', $nome='',
$codiceFiscale='', $stato='')
```

Funzione per recuperare array con tutti gli utenti, è possibile passare un *array* di ID nel parametro `id utente`, oppure singolarmente i vari campi `username` `cognome` `nome` o `codice fiscale`.

```
getUtentiApplicazione ($id_applicazione, $not=0)
```

Ritorna l'elenco degli utenti abilitati ad una determinata applicazione (con almeno un'abilitazione sull'applicazione) è possibile passare un secondo parametro all'applicazione (`$not`), con un valore diverso da 0 o *false*, per ottenere gli utenti che non sono abilitati.

```
getGruppiUtente($id_utente)
```

Passando a questa funzione l'id di un utente, la funziona restituisce un array con gli identificativi dei gruppi a cui l'utente appartiene, in caso non appartenga a nessun gruppo la funziona ritornerà *false*.

```
getApplicazione($programma=true, $dataset="")
```

Funzione che individua l'applicazione aperta sul client analizzando ricorsivamente il *path* da cui è stata effettuata la richiesta.

```
getAbilitazioni($id_utente='', $id_applicazione='', $variabile='',
$valore='')
```

Questa funzione restituisce le abilitazioni dalla tabella relativa, in \$id_utente è possibile passare singolarmente un ID utente o un ID gruppo oppure anche solo uno degli altri parametri singolarmente.

```
checkLoginAD($username, $password, $tipoAutenticazione)
```

Funzione per l'autenticazione dell'utente su dominio *active directory*, la funzione restituisce un *dataset* contenente l'esito dell'autenticazione (*true* o *false*), il messaggio di risposta da parte del server LDAP, e una variabile generata in base al messaggio restituito.

Nel caso in cui l'autenticazione non vada a buon fine la funzione suggerisce mediante quest'ultima variabile quale pagina deve essere caricata, ad esempio nuovamente il login o la schermata di cambio password nel caso in cui le credenziali utente siano scadute.

```
checkCambioPassword($username, $passwordattuale='', $password,
$conferma, $type='')
```

Funzione che implementa tutti i controlli per il cambio password, verifica che la password rispetti le regole di complessità imposte e cioè: lunghezza minima otto e massima duecentocinquantacinque caratteri, almeno una lettera maiuscola, almeno una lettera minuscola, almeno un numero, nessun carattere speciale è consentito, non può contenere più di tre lettere consecutive presenti nello username e verifica inoltre che le ultime 3 password non coincidano a quella scelta come nuova.

La password è salvata sul database codificando con l'algoritmo MD5 la stringa id_utente|password, essendo MD5 un algoritmo di codifica *one-way*, per verificare lo storico delle password è necessario codificare secondo lo stesso schema anche la nuova password e confrontare le stringhe codificate.

```
abilitato($variabile='', $valore='')
```

E' una funzione comoda per gli sviluppatori di applicazioni esterne che sfrutta le variabili di ambiente del server (per ottenere l'id_applicazione), quelle di sessione (dove viene salvato l'id_utente) e le funzioni menzionate in precedenza per verificare che un utente sia abilitato o meno (*true* o *false*) semplicemente passando al programma i parametri variabile e valore, come salvati all'interno del database.

Questo permette con un semplice *if* di includere o escludere alla visualizzazione una porzione di pagina all'utente utilizzatore.

3.1.2 Logica di autenticazione

Come detto in precedenza, oltre ad includere tutte le funzioni implementate nella libreria vista in precedenza, nel file `irst.base.php` è codificata una prima parte della logica con cui il programma procede alla verifica delle credenziali e, in base al ritorno ottenuto decide quale sia il passo successivo da presentare all'utente.

Estratto del file `irst.base.php`:

```
<?php
//esclusione dei controlli se la pagina è richiamata utilizzando ajax
if ($_SERVER['HTTP_X_REQUESTED_WITH'] != 'XMLHttpRequest'){
    //identificazione dell'applicazione
    $applicazione = getApplicazione("all");
    //set della sessione e della variabile applicazione
    session_start();
    $_SESSION['irst_id_applicazione'] = $applicazione['id_applicazione'];
    //inclusione delle librerie jquery, jqueryUI e css relativi
    include_once('jquery.inc.php');
    //se l'app è codificata su DB
    if ($applicazione){
        //se l'utente non è settato tra le variabili di sessione
        if (!isset($_SESSION['irst_id_utente'])) {
            //viene mostrata la pagina di login
            include_once($_SERVER.'/lib/irst/pages/loginProcess.php');
            die();
        }
        //vengono controllati i permessi dell'utente sull'applicazione
        includendo la barra applicazione
        else include_once($_SERVER.'/lib/irst/pages/header.php');
        // Se l'applicazione non è codificata
    else{
        //viene presentato un errore a schermo
        include_once($_SERVER.'/lib/irst/pages/header_errore.php');
    }
    // altrimenti si procede con l'applicazione
}
```

In questo file viene verificato che il programma visualizzato sia inserito correttamente nel *database* applicazioni e che richieda l'autenticazione, nel caso in cui non fosse presente verrà mostrato una pagina di errore.

Successivamente viene verificata la variabile di sessione `id_utente`, nel caso in cui non fosse presente, all'utente viene proposta la pagina di autenticazione (che si assume il compito di impostare la variabile `id_utente` una volta verificate le credenziali), viceversa il programma andrebbe ad includere la "barra applicazione" (*paragrafo 2.3.1*) e caricare di conseguenza l'applicazione.

3.1.3 Pagina di login e recupero password

La logica di funzionamento è contenuta all'interno di un singolo file: *loginProcess.php*.

Questa pagina include tutto il codice html e javascript per la visualizzazione del login, del cambio password e del reset dell'account in caso di credenziali perse.

Grazie all'utilizzo di diverse chiamate AJAX al server, la pagina non necessita di effettuare alcun *refresh*, rendendo l'interazione più naturale, e grazie all'impiego della variabile javascript `document.location.search` è anche possibile richiamare direttamente le varie schermate che compongono la pagina.

La pagina di *login* si presenta con una facciata minimalista, rispetto al progetto iniziale è stato tolto il logo per alleggerirne ancora di più l'aspetto (*figura3.1*).

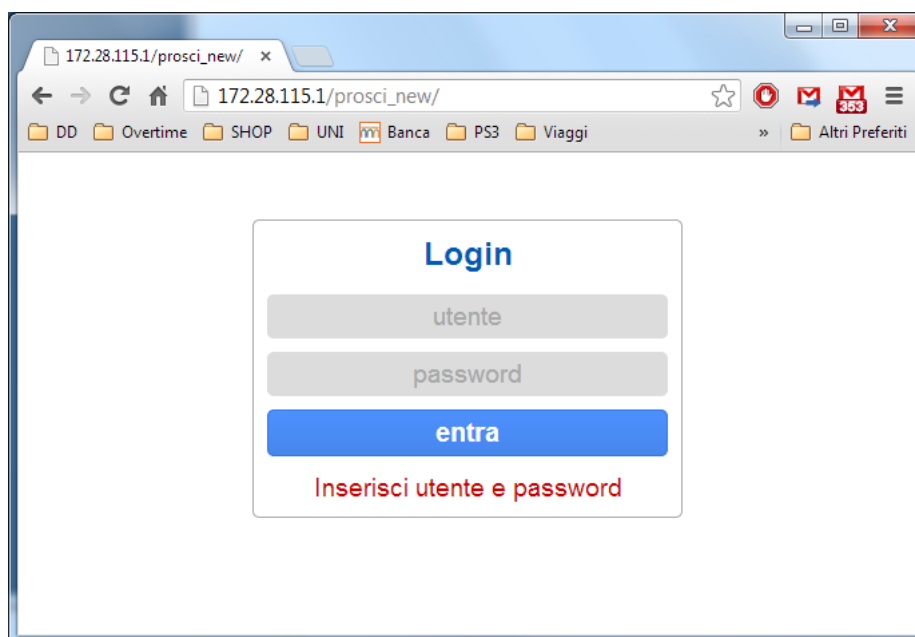


Figura 3.1 - Login IRST

Nella pagina vengono presentati due campi: utente e password e un pulsante per effettuare il login.

Sotto il pulsante è presente uno spazio dove possono comparire una serie di messaggi di allerta, nel caso uno dei due campi non sia stato compilato verrà ricordato all'utente di farlo con la scritta "inserisci utente e password", nel caso in cui, come in *figura 3.2*, siano state inserite credenziali non corrette il messaggio ci propone anche un link per richiedere il reset delle credenziali.

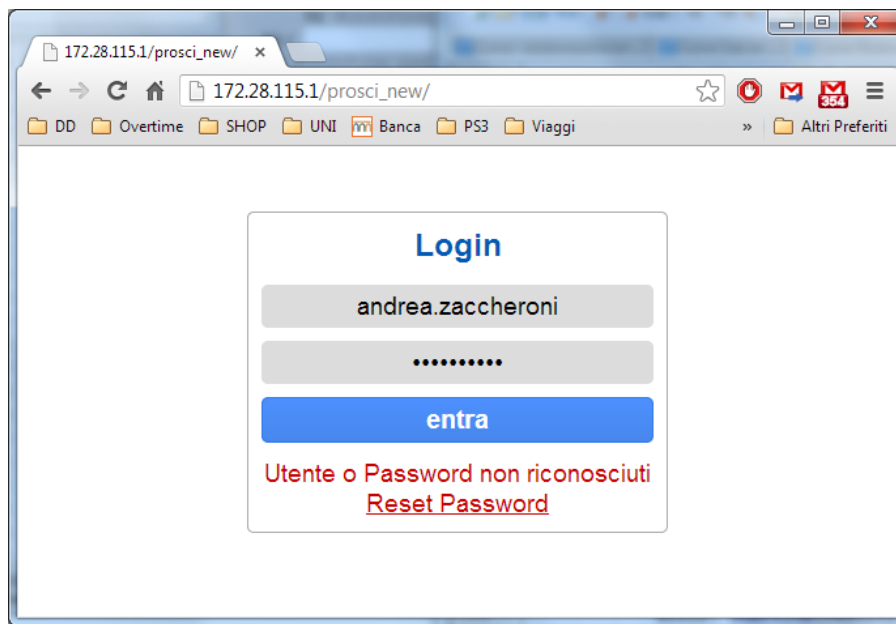


Figura 3.2 - Utente o password non riconosciuti

Questi messaggi di errore vengono restituiti da pagine PHP chiamate sfruttando AJAX, che con una serie di tentativi, utilizzando le funzioni messe a disposizione dalla libreria presentata in precedenza, verificano l'utente, nel caso in cui esistesse individuano tramite la variabile *tipo_utente* se si tratta di un utente di dominio oppure di un utente gestito dal database locale e attivano una serie di procedure di conseguenza fino a stabilire se le credenziali fornite sono corrette, incorrette o scadute.

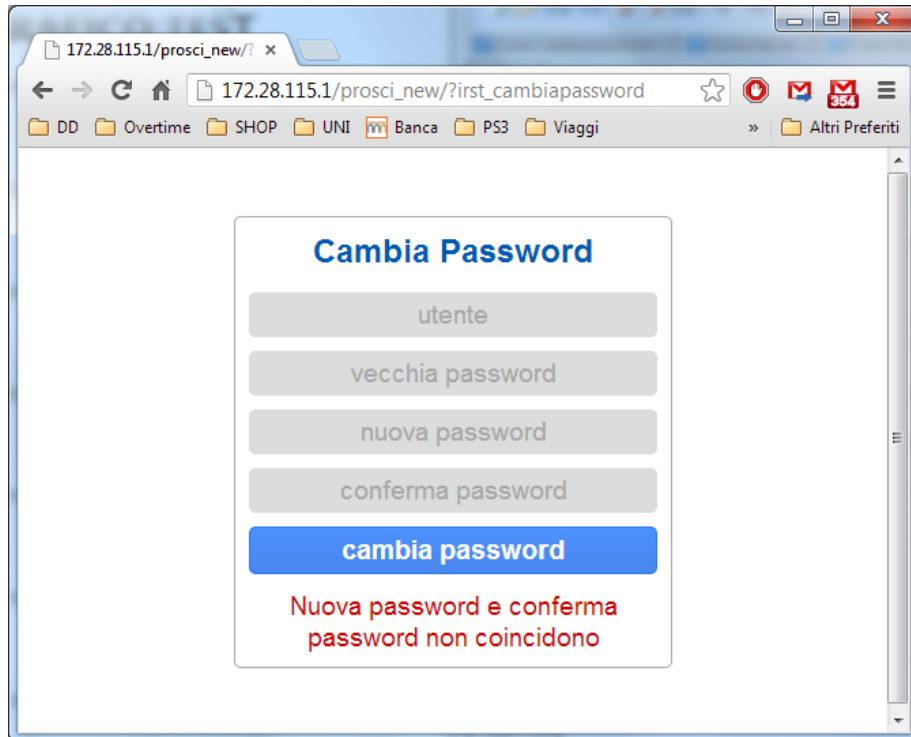


Figura 3.3 - Modifica Password

Nel caso di credenziali corrette ovviamente il software permette di accedere all'applicazione richiesta.

Se invece le credenziali dell'utente dovessero essere scadute verrà presentata la pagina di cambi password (figura 3.3)

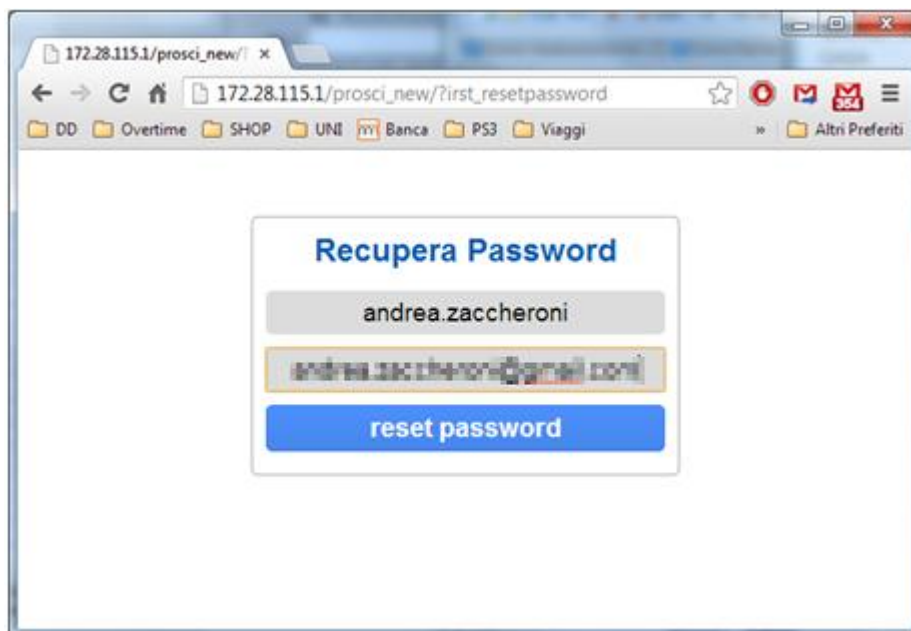


Figura 3.4 - Recupera Password dimenticata

In caso di credenziali incorrette segnala il problema con il messaggio volutamente generico "utente o password non riconosciuti" e il link per il reset delle credenziali (schermata mostrata in *figura 3.4*).

Come per la parte di autenticazione anche la parte di cambio password è gestita da una pagina PHP chiamata tramite AJAX:

Esempio di chiamata AJAX con l'utilizzo dell'oggetto `$.ajax` di jQuery:

```
$.ajax({
    url: "<?=$server?>/lib/irst/pages/new_reset_password.php",
    data: {username:$('#username').val(),email:$('#email').val()},
    type: "POST",
    dataType: "json",
    success: function(result){
        if (result.success){
            //ELABORAZIONE DEL RISULTATO ASPETTATO
            ...
        }
        //comparsa del messaggio di warning per l'utente
        //all'interno della pagina
        else if(!result.success){
            if (result.code == 'change_pwd'){
                $('#irstMsg').html(result.msg)};
                ...
            }
            //in caso di risposta non attesa dal server
            //alert del codice html ricevuto per il debug
            else alert(JSON.stringify(result));
        }
    },
    // in caso di errore nella comunicazione col server
    //alert dell'errore per il debug
    error: function(result){ alert(JSON.stringify(result))}
});
```

Come si può notare dal codice riportato la comunicazione tra client e server avviene mediante l'utilizzo di strutture dati JSON.

Nel parametro *url* viene passata la pagina da richiamare sul server, in data i parametri da passare all'url richiesto con il metodo indicato nel parametro *type* (*get* o *post*), in *dataType* viene specificato il tipo di messaggio di risposta che ci si aspetta dal server (testo, html, xml o json).

Discorso a parte per *success* e *error*, qui è possibile specificare una funzione, detta di *callback*, da eseguire nel caso in cui si riesca a comunicare col server (*success*) o meno (*error*).

In particolare nel caso di login il client invia al server un oggetto di questo tipo:

```
{
  irst_username: "username",
  irst_password: "password"
}
```

e riceve un oggetto JSON così strutturato:

```
{
  result: true,
  code: "codice",
  message: "messaggio di testo"
}
```

3.1.4 Barra applicazione

Nel caso in cui l'autenticazione vada a buon fine le pagine php procedono a impostare le variabili di sessione necessarie, tra cui *user_id*.

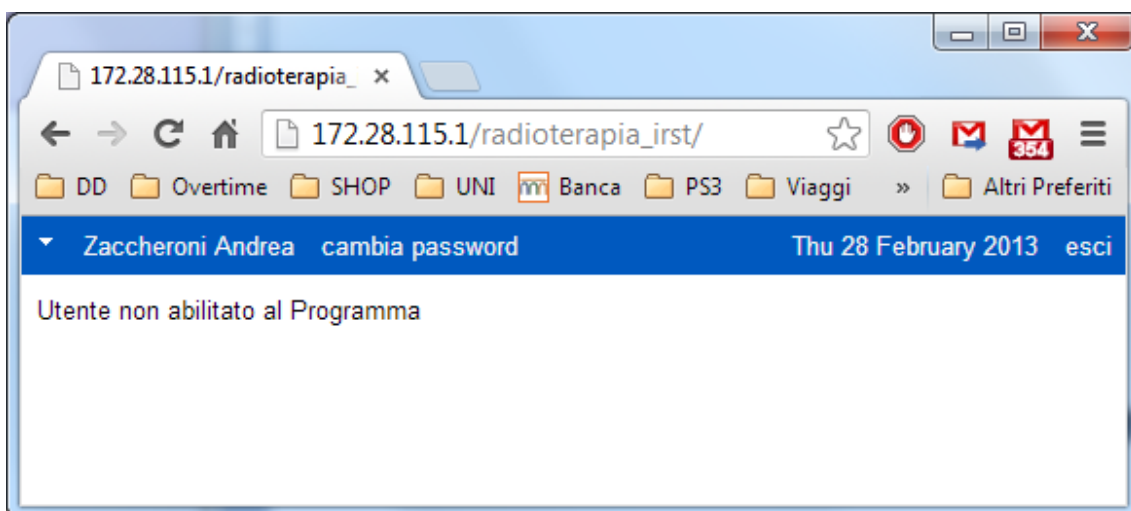


Figura 3.5 - Barra Applicazione Utente non abilitato al programma

Prima di procedere al caricamento dell'applicazione viene caricata la pagina relativa alla barra applicazione, questa pagina verifica che l'utente abbia una qualsiasi abilitazione al programma richiesto e in caso contrario visualizza un messaggio di errore.

In ogni caso viene presentata una barra di colore azzurro nella parte alta dello schermo come visibile in *figura 3.5*.

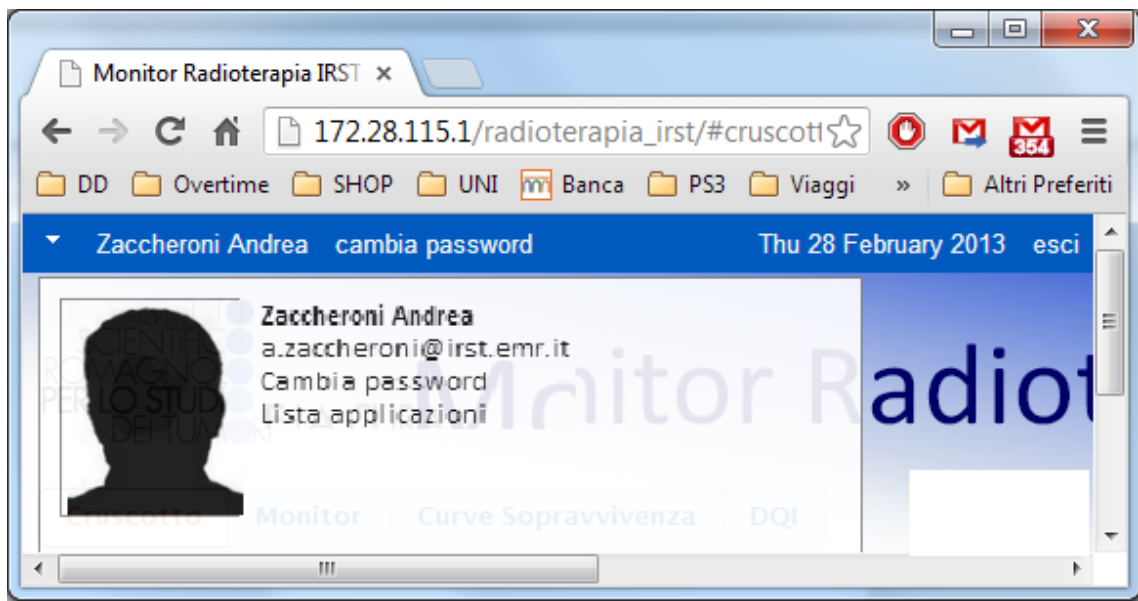


Figura 3.6 - Tendina opzioni utente

Sulla barra troviamo il nome e il cognome dell'utente, un pulsante rapido per il cambio della password, che verrà sostituito nella prossima versione del software con il menu per poter variare i propri permessi all'interno della pagina, la data, al momento in inglese, e un pulsante esci per effettuare il *logout* dal sito.

Andando a cliccare sul nome e cognome comparirà una finestra con alcuni dei dati utente e un paio di link, per il momento cambia password e lista applicazioni, per la gestione del proprio account come mostrato in *figura 3.6*.

3.2 Modulo statistiche Amministrative

Allo stato attuale le informazioni relative a questo modulo non sono state organizzate all'interno di questa sezione del portale che rimane in fase di sviluppo.

Sebbene i dati siano pronti e le estrazioni effettuate, non è ancora chiaro alla direzione quali siano le informazioni da visualizzare all'interno di questa sezione.

Di seguito verranno mostrate le aggregazioni di dati generati partendo dalla vista messa a disposizione sul database del software LOG80.

Tabella generata aggregando i dati annuali relativi all'unità di radioterapia IRST:

Periodo	Prestazioni	Nr.Pazienti		Nr.Cartelle		Nr.Prestazioni	
		Nr.	%	Nr.	%	Nr.	%
Anno 2011	Complessivo	2204	100,00	43092	100,00	54310	100,00
Anno 2011	AVR	1714	77,77	35463	82,30	44171	81,33
Anno 2011	Extra RER	329	14,93	5304	12,31	7150	13,17
Anno 2011	RER Extra AVR	161	7,30	2325	5,40	2989	5,50
Anno 2011	VISITA RADIOTERAPICA PER CONTROLLO	1641	74,46	6439	14,94	6443	11,86
Anno 2011	VISITA RADIOTERAPICA PRETRATTAMENTO	1590	72,14	1828	4,24	1828	3,37
Anno 2011	INDIVIDUAZIONE DEL VOLUME BERSAGLIO E SIM	1244	56,44	1473	3,42	1473	2,71
Anno 2011	SISTEMA DI IMMOBILIZZAZIONE PERSONALIZZATO	1232	55,90	1456	3,38	1460	2,69
Anno 2011	DEFINIZIONE TARGET ED ORGANI CRITICI PER RADI	1098	49,82	1292	3,00	1545	2,84
Anno 2011	TC DI SET UP	866	39,29	9411	21,84	9605	17,69
Anno 2011	STUDIO FISICO DOSIMETRICO CON ELABORATORE SI	776	35,21	899	2,09	942	1,73
Anno 2011	TOMOTERAPIA	636	28,86	6190	14,36	12935	23,82
Anno 2011	STUDIO FISICO DOSIMETRICO CON ELABORATORE SI	487	22,10	516	1,20	537	0,99
Anno 2011	SCHERMATURA PERSONALIZZATA (intero trattamento)	452	20,51	479	1,11	1929	3,55
Anno 2011	TELETERAPIA CON ACCEL. LINEARE CON CAMPI MULTIFASICI	326	14,79	6176	14,33	7196	13,25
Anno 2011	CONTROLLO SET UP INIZIALE per 1a SEDUTA	308	13,97	973	2,26	1970	3,63
Anno 2011	INIEZIONE DI m.d.c. PER SIMULAZIONE	239	10,84	259	0,60	259	0,48
Anno 2011	TELETERAPIA CON ACCEL. LINEARE CON CAMPI MULTIFASICI	169	7,67	3609	8,38	3701	6,81
Anno 2011	DOSIMETRIA IN VIVO	138	6,26	168	0,39	423	0,78

Figura 3.7 - Analisi dati aggregati prestazioni radioterapia

In questa estrazione possiamo vedere il dettaglio per l'anno 2011 delle prestazioni effettuate in radioterapia IRST (i numeri sono stati alterati), secondo diversi raggruppamenti (figura 3.7).

Nella prima riga in alto i dati vengono semplicemente sommati dando un prospetto complessivo dell'attività svolta in radioterapia,

Nelle righe successive vengono raggruppate le prestazioni secondo la provenienza del paziente, con AVR si indica un paziente che proviene dalle AUSL di Forlì, Cesena, Rimini, Ravenna o Imola, con RER extra AVR i pazienti provenienti dalla regione Emilia-Romagna provenienti dalle AUSL al di fuori di area vasta, e con extra RER tutti i pazienti provenienti da altre regioni o dall'estero.

Nelle righe successive viene dettagliato invece il numero di pazienti e prestazioni effettuate raggruppate per ogni singolo tipo di prestazione.

Da questa tabella, ottenuta eseguendo una serie di *query* sulla vista a nostra disposizione è possibile tenere monitorato, per fasce di tempo diverse che includono l'anno, il semestre fino al mese in corso, la situazione riguardante le prestazioni effettuate e da qui generare una serie di indicatori grafici.

Un'altra aggregazione è quella realizzata aggregando i dati per fascia di età:

Tipo Prestazione	Fascia	Nr.Pazienti	Nr.Prestazioni	Prestazioni / paziente
A - Prima Visita	25 - 29	1	1	1
A - Prima Visita	30 - 34	6	6	1
A - Prima Visita	35 - 39	5	5	1
A - Prima Visita	40 - 44	13	13	1
A - Prima Visita	45 - 49	16	16	1
A - Prima Visita	50 - 54	17	18	1,06
A - Prima Visita	55 - 59	29	31	1,07
A - Prima Visita	60 - 64	36	36	1
A - Prima Visita	65 - 69	45	48	1,07
A - Prima Visita	70 - 74	39	40	1,03
A - Prima Visita	75 - 79	42	43	1,02
A - Prima Visita	80 - 84	15	15	1
A - Prima Visita	85 - 89	9	10	1,11
A - Prima Visita	90 - 94	2	2	1
B - TAC	25 - 29	2	5	2,5
B - TAC	30 - 34	2	3	1,5
B - TAC	35 - 39	6	13	2,17
B - TAC	40 - 44	15	40	2,67
B - TAC	45 - 49	20	78	3,9
B - TAC	50 - 54	24	57	2,38
B - TAC	55 - 59	32	93	2,91
B - TAC	60 - 64	46	84	1,83

Figura 3.8 - Analisi dati aggregati prestazioni radioterapia 2

Partendo da questi dati si potrebbe realizzare un istogramma o un digramma a torta per mostrare a quale fascia di età corrisponde il maggior numero di prestazioni somministrate.

3.3 Modulo statistiche di Reparto

In questo modulo vengono presentati i livelli di occupazioni dei singoli ambulatori e delle diagnostiche con un istogramma che rappresenta giorno per giorno mediante l'uso di colonne divise in settori di colore diverso i posti (chiamati slot dal gergo utilizzato da Mosaiq) prenotati, le visite/terapie eseguite, i posti liberi, i fermo macchina o le indisposizioni dei pazienti.

Tutti i componenti dell'interfaccia grafica a meno dei grafici e delle tabelle, sono componenti presi dalle librerie jQueryUI, in particolare si è utilizzato il *widget button* per il menu principale (nella parte alta), i widget tabs per creare il contenitore principale e quello per il menù di sinistra.

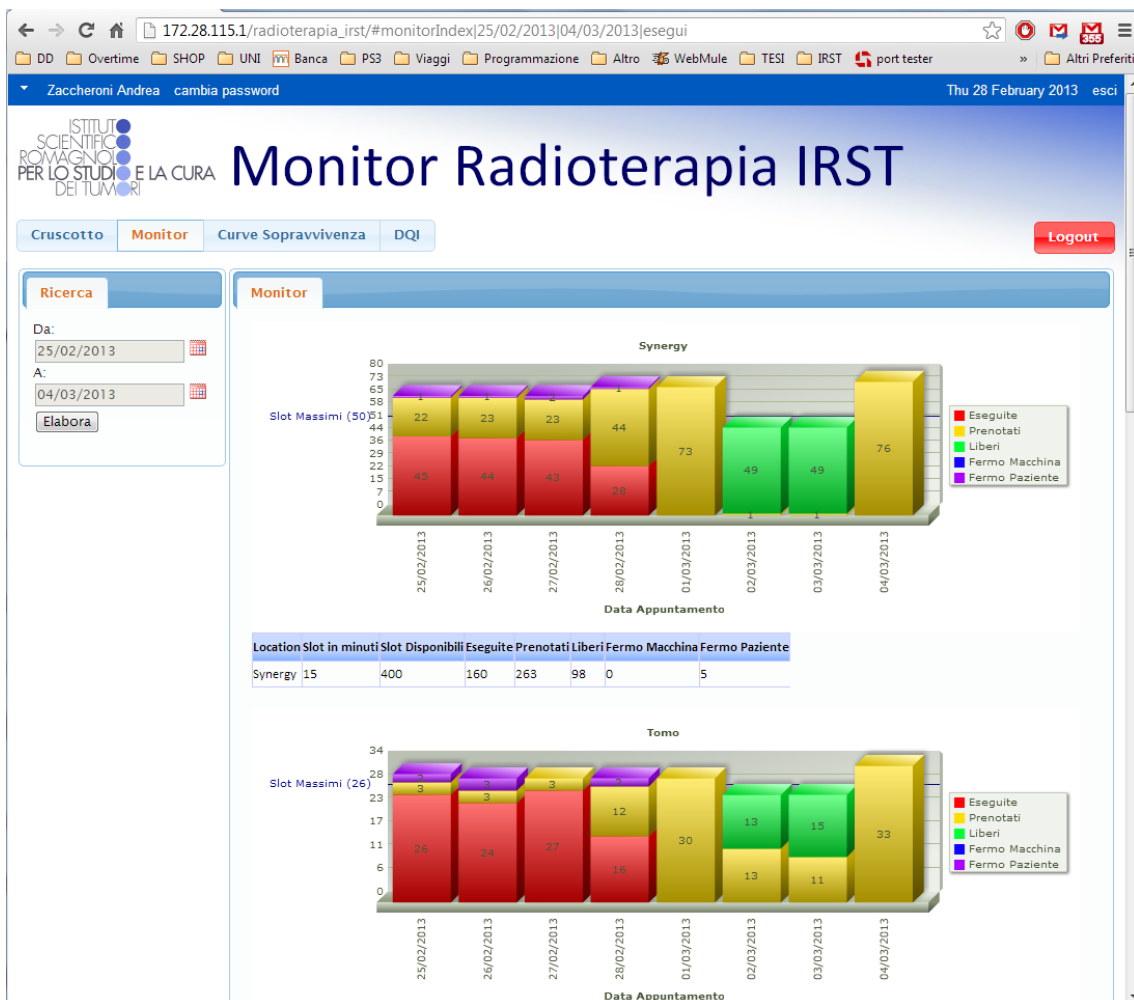


Figura 3.9 - Monito ambulatori e diagnostiche

All'apertura della pagina "monitor"(figura 3.9), vengono automaticamente caricati i dati relativi alla settimana in corso.

Mentre il software carica i dati estraendoli direttamente dalla vista *vw_schedule* sul *database* del *software* Mosaik viene bloccata l'interazione da parte dell'utente con una schermata di caricamento come mostrato in figura 3.10.

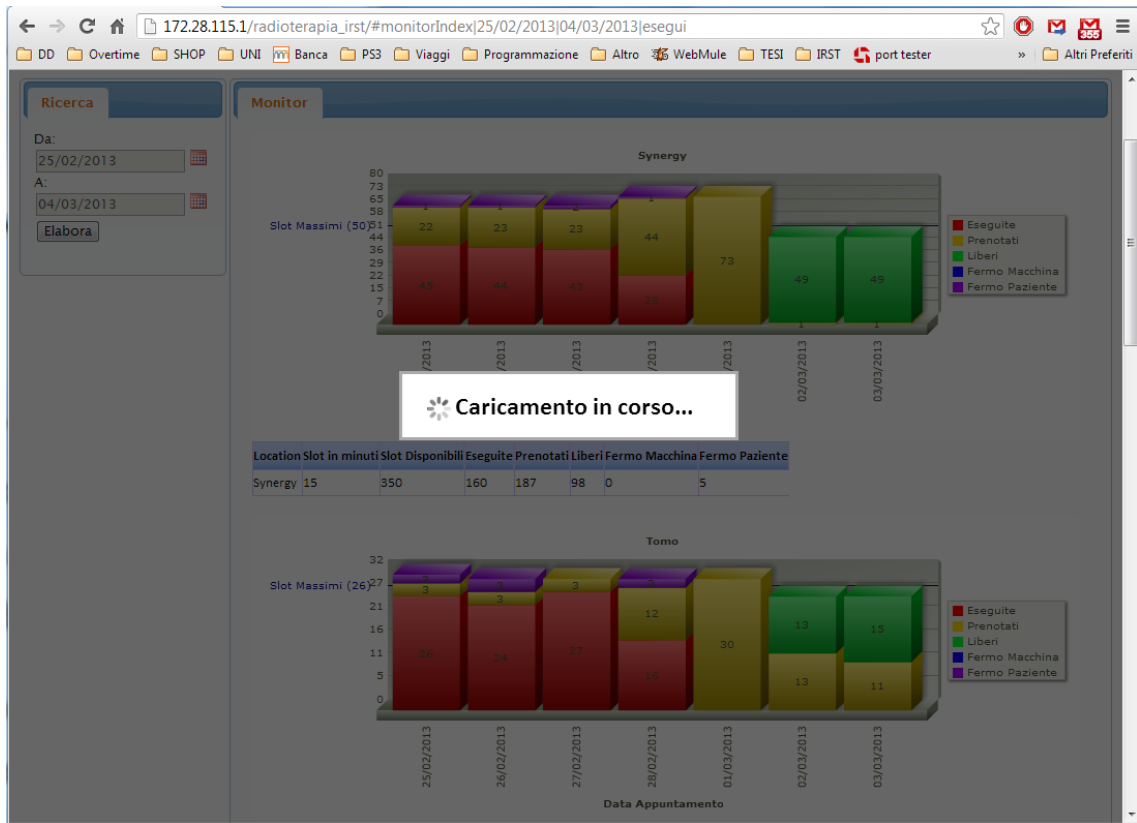


Figura 3.10 - Schermata di attesa

Per ottenere questo effetto si è utilizzato un *plugin* jQuery, BlockUI.

Grazie a questo *plugin* è sufficiente chiamare la funzione `$.blockUI()` prima della chiamata AJAX e la funzione `$.unblockUI()` sulla sua callback [JUI13].

Esempio di chiamata AJAX con blocco dell'interfaccia:

```
$.blockUI();
$.ajax({
  url: "test.html",
  dataType: "html",
}).done(function() {
  $.unblockUI();
});
```

Una volta caricati i dati dal database questi vengono visualizzati sotto forma di grafico sfruttando le librerie FusionCharts [FUS13].

Per generare i grafici come li possiamo vedere nelle figure 3.9 e 3.11 è necessario produrre e sottoporre al *plugin* una struttura dati XML di questo tipo:

```
<graph parametri...>
  <categories>
    <category name='01/02/2013' showName='1' />
  </categories>
  <dataset seriesName='Eseguite' color='FF0000'>
    <set value='45' alpha='100' link='JS:apriGiornata(...);'>
    </set>
  </dataset>
  <dataset seriesName='Prenotati' color='FFDD00'>
    <set value='12' alpha='100' link=JS:apriGiornata(...);'>
    </set><set alpha='100'>
    </set>
  </dataset>
  <dataset seriesName='Liberi' color='00FF33'>
    <set alpha='100'>
    </set>
  </dataset>
  <dataset seriesName='Fermo Macchina' color='1100FF'>
    <set alpha='100'>
    </set>
  </dataset>
  <dataset seriesName='Fermo Paziente' color='AA00FF'>
    <set value='2' alpha='100' link=JS:apriGiornata(...);'>
    </set>
  </dataset>
  <trendlines>
    <line startValue='50'color='000088' displayValue='Slot
Massimi (50)' showOnTop='1' />
  </trendlines>
  <legend/>
</graph>
```

All'interno della struttura dati notiamo in particolar modo, i *tag category* dove sono definite le singole colonne e i *dataset* che definisco i segmenti della singola colonna e contengono come parametro il valore dello stesso.

All'interno di quest'ultimi è possibile passare come parametro una funzione javascript, utilizzata per mostrare il dettaglio della giornata come mostrato in *figura 3.11* al click dell'utente.

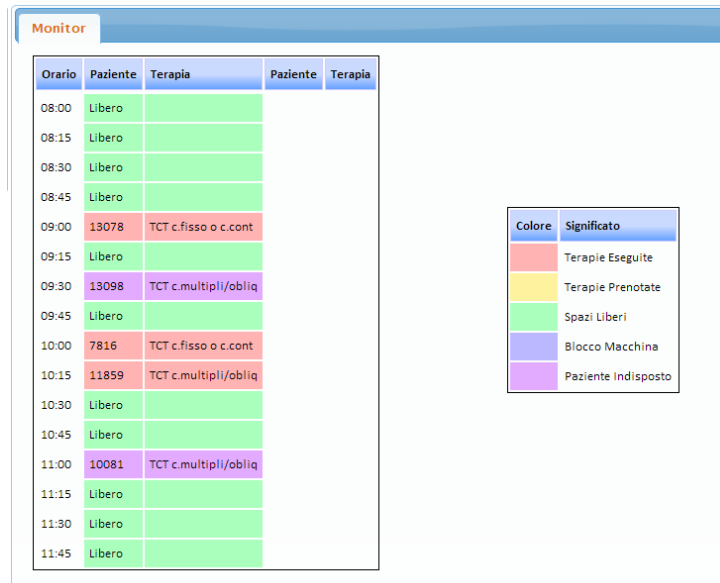


Figura 3.11 - Dettaglio appuntamenti diagnostica

Andando ad utilizzare i controlli *datepicker* è possibile effettuare una nuova ricerca e selezionare un intervallo di tempo fino a 90 giorni.

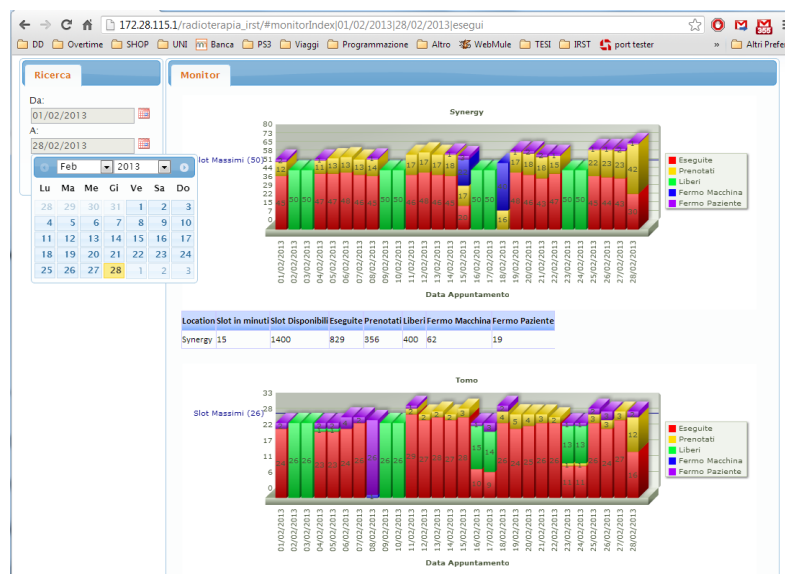


Figura 3.12 - Datepicker e intervallo di tempo maggiore

3.4 Modulo Curve di Sopravvivenza

Questo modulo è il più complesso dal punto di vista dell'interfaccia utente e della realizzazione, infatti convergono in esso una serie di competenze in ambiti diversi, dalla medicina, alla statistica fino alla programmazione web.

Nel paragrafo analizzeremo le funzioni PHP utili a reperire le informazioni e generare le strutture dati per la comunicazione tra i vari elementi del software, analizzeremo l'interfaccia web e tutti gli elementi che la compongono.

3.4.1 Interfaccia di selezione dei casi da analizzare

Il menu si presenta come lo possiamo vedere in *figura 3.13*, cliccando sui primi due campi verrà visualizzato come nella pagina precedente l'elemento *datepicker* per la scelta della data.

Il campo codice paziente serve principalmente ai medici per individuare tutti i casi visti nell'arco di un determinato anno, infatti i codici paziente sono preceduti dall'anno in cui si è effettuata la prima visita in reparto.

Grazie all'utilizzo della *wildcard **, utilizzabile anche in tutti i campi seguenti, è infatti possibile indicare solo parzialmente il valore da ricercare.

Per i campi codice patologia, origine diagnosi, codice istologia e intento terapia è stato sviluppato un controllo ad hoc che sfrutta il *widget autocomplete* di jQueryUI, per il suggerimento del campo da ricercare, aggiungendo la possibilità di moltiplicare il numero di valori per il singolo campo andando a fare *click* sul pulsante "+" come mostrato nella *figura 3.14*.

Figura 3.13 - Menu curve di sopravvivenza

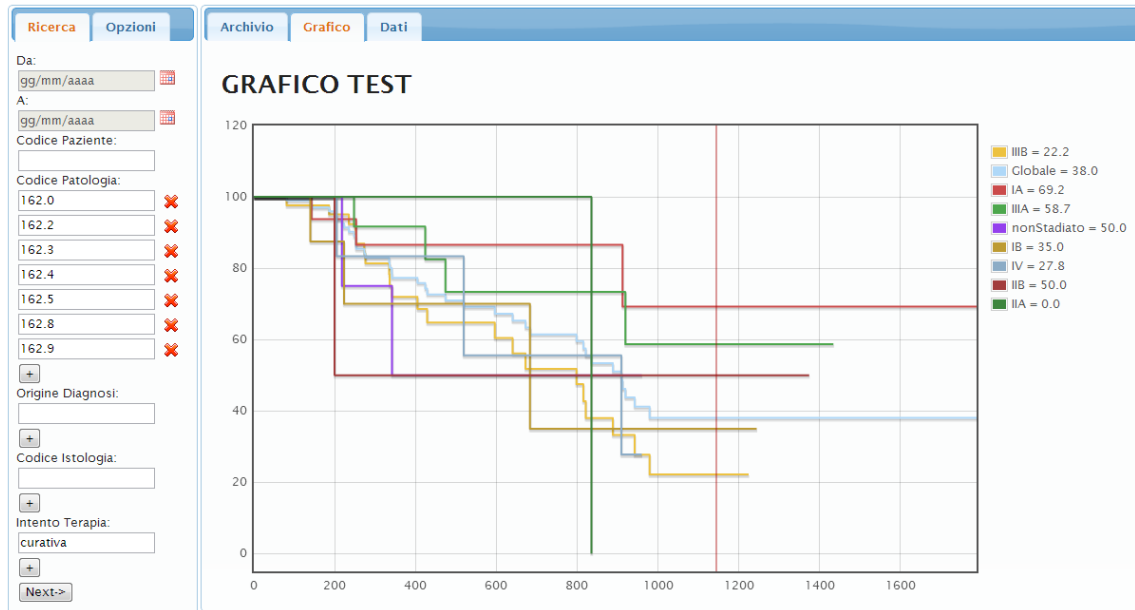


Figura 3.14 - Esempio di estrazione

In questa schermata si può vedere come è possibile moltiplicare i campi, nell'esempio il codice di patologia.

Analizziamo il codice javascript utilizzato per generare i campi codice di patologia:

```
function aggiungiAuto(classe, tipo) {
var clonedEl = $('.'+classe+':last').clone();
$(clonedEl).find('input:text').val('');
clonedEl.insertAfter('.'+classe+':last');
$('.'+classe).find('img[src*="delete"]').show();
var source = $('.' + classe + ':first').children('input:text')
.autocomplete( "option", "source" );

    switch(tipo) {
        case "patologia":
            initAutocompletePat(source);
            break;
        case "istologia":
            initAutocompleteIst(source);
            break;
        ...
        default:
            alert('errore');
            break;
    }
}
```

```
}  
function initAutocompletePat(patologie){  
  $(' .curvepatologia').each(function(i, el) {  
    el = $(el);  
    el.autocomplete({  
      minLength: 0,  
      source: patologie,  
      focus: function( event, ui ) {  
        $(this).val( ui.item.Topography );  
        return false;  
      },  
      select: function( event, ui ) {  
        $(this).val(ui.item.Topography);  
        return false;  
      }  
    })  
    .data("autocomplete")._renderItem = function( ul, item ) {  
      return $( "<li></li>" ).data("item.autocomplete", item )  
    };  
  }  
}
```

Le funzioni che formano il controllo sono due, la prima serve principalmente a duplicare il campo e disegnarlo all'interno della pagina, individuare il tipo di controllo e personalizzare la sorgente dei dati per *l'autocomplete*, la seconda invece effettivamente inizializza il *widget autocomplete* sul nuovo elemento.

3.4.2 Interfaccia di post elaborazione

Una volta inviati i dati relativi al filtro, l'interfaccia si sposta automaticamente su un secondo menu che permette di effettuare raggruppamenti dei dati al fine di disegnare curve relative a stadi patologici diversi (*figura 3.15*).

Viene inizialmente presentato un solo campo, cliccando su di esso sarà possibile impostare un'etichetta per il raggruppamento, sotto di esso comparirà una lista di tutti i tipo di stadio estratti, con affianco una *checkbox*, selezionando una o più *checkbox* si imposta il raggruppamento per la curva.

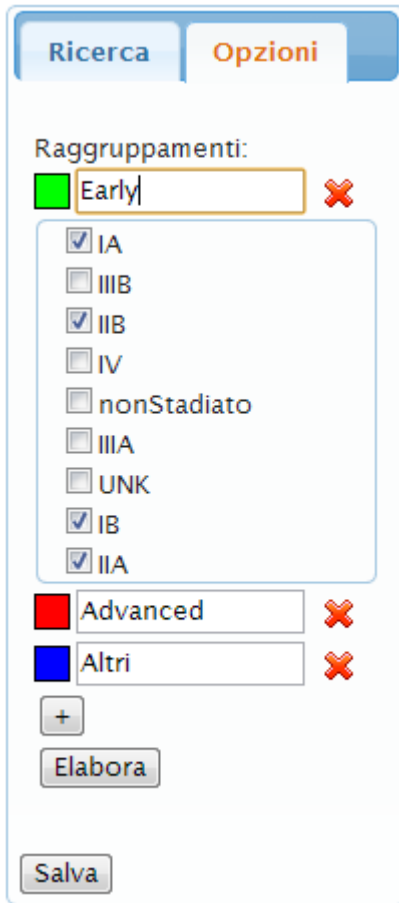


Figura 3.15 - Menu post-elaborazione

E' ovviamente possibile aggiungere un nuovo raggruppamento facendo *click* sul pulsante "+" e selezionare criteri diversi di raggruppamento.

Infine per ogni curva è possibile selezionare un colore andando a sceglierlo da un controllo chiamato *colorpicker* anche questo reso disponibile come plugin di jQuery che consente di selezionare un colore da una tavolozza con un semplice *click* del mouse [COL11].

Oltre al raggruppamento dei dati attraverso questo menù, selezionando la tabella "Dati" nel contenitore principale (quello dove viene rappresentato il grafico), è possibile avere la visione completa di tutti i casi che contribuiscono a formare la curva.

Per ogni caso vengono visualizzati una serie di valori utili al medico per identificare il paziente e il tipo di malattia oltre ad un campo che viene generato in base ai parametri scelti in post elaborazione e che indica a quali delle curve create il dato afferisce.

Questo permette al medico di rendersi conto se il filtro applicato con il primo "form" o il raggruppamento effettuato con il secondo siano corretti.

CC	Data_Diagnosi	Origine_Diagnosi	Stadio_Clinico	Stadio_Patologico	Codice_Diagnosi	codice_Istologia	OS_stato	OS_gg	Ultima_Visita	Gruppi	Abilitato
11459	17/01/2011	LUNG 162	IIIB		162.3	80703.00	0	184	19	(Advanced)	✓
20020443	14/05/2010	LUNG 162	IIIB		162.5	81403.00	0	405	20	(Advanced)	✓
20030138	15/10/2010	LUNG 162	IIIA		162.5	81403.00	0	210	21	(Advanced)	✗
20050265	18/10/2010	LUNG 162	IIIB		162.3		1	498		(Advanced)	✓
20060003	15/12/2009	LUNG 162	IV		162.9	80463.00	1	300		(Advanced)	✗
20060302	22/10/2009	LUNG 162	IIIA		162.5	82401.00	0	158	35	(Advanced)	✓
20070300	06/05/2010	LUNG 162	IIIB		162.3	81403.00	1	251		(Advanced)	✓
20070359	14/05/2007	LUNG 162	IV		162.3	80123.00	1	258		(Advanced)	✗
20070421	15/12/2009	LUNG 162	IA		162.5	80703.00	1	460		(Early)	✓
20070560	24/08/2007	LUNG 162	IB		162.3	81403.00	1	222		(Early)	✓
20070616	22/02/2007	LUNG 162	IB		162.3	81403.00	0	1243	31	(Early)	✗
20070623	06/09/2007	LUNG 162	IIIB		162.5	80413.00	0	1098	29	(Advanced)	✗
20070658	14/08/2007	LUNG 162	IIIA		162.3	80703.00	1	234		(Advanced)	✗
20070685	18/05/2007	LUNG 162	IIIB		162.5	80703.00	1	241		(Advanced)	✗
20070698	01/06/2007	LUNG 162			162.5	80723.00	1	217		(Altri)	✓
20070709	28/06/2007	LUNG 162		IIIB	162.5	82403.00	0	277	59	(Advanced)	✓
20070716	19/06/2007	LUNG 162	IIA		162.4		1	834		(Early)	✓

Figura 3.16 - Dati curve

Semplicemente cliccando sul campo "abilitato" su ogni singola riga è possibile escludere o includere nuovamente quel determinato caso dall'elaborazione del grafico.

3.4.3 Visualizzazione grafica interattiva dei risultati

Per quanto riguarda il grafico, mostrato in *figura 3.17*, si è utilizzato come detto in precedenza il *plugin jQuery "Flot"*, andandolo a personalizzare in una serie di funzionalità.

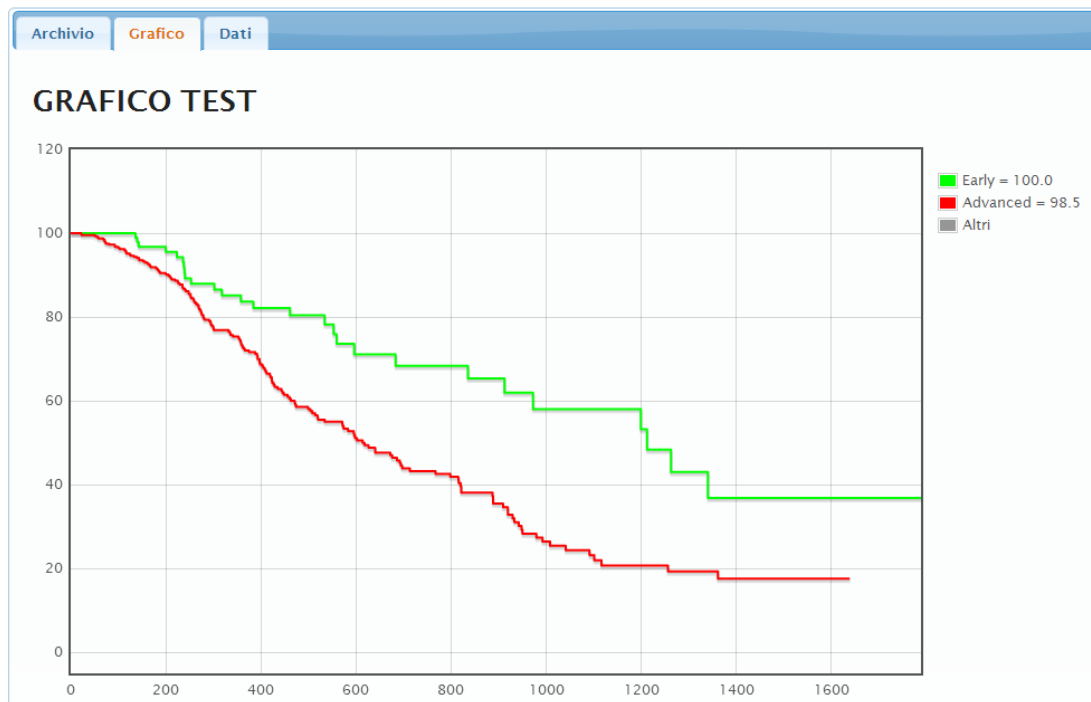


Figura 3.17 - Grafico con raggruppamento

Cliccando nella legenda, sul colore di una serie, è possibile disabilitare o riabilitare la visualizzazione della curva corrispondente.

Andando a muovere il mouse sull'immagine, viene visualizzata una linea verticale che interseca le curve in un punto il cui valore, è indicato in tempo reale a fianco del nome della curva all'interno della legenda.

Per disegnare all'interno del browser Flot utilizza l'elemento *HTML5 canvas* [LAU13], questo elemento non è disponibile sui browser Microsoft Internet Explorer nelle versioni precedenti alla nove

Per questo motivo quindi, è stato necessario includere all'interno del progetto una libreria aggiuntiva scritta in javascript, *ExplorerCanvas*, che aggiunge queste funzionalità andando a sfruttare il *VML redererer* di Internet Explorer, invece presente in modo nativo[EXP09].

3.4.4 Strutture dati XML e JSON

Il primo filtro genera una richiesta tramite AJAX ad una pagina PHP che estrae i dati dalla vista direttamente dal database di Mosaiq.

I parametri di ricerca vengono passati alla pagina come un oggetto JSON così definito:

```
{
  "patologie":["162.*"],
  "istologie":[""],
  "datai":"",
  "dataf":"",
  "cartelle":""
}
```

Viene utilizzato un altro oggetto JSON per i parametri di raggruppamento impostati nella seconda parte del menu di generazione del grafico:

```
[
  {
    "name":"Early",
    "color":"rgb(0,255,0)",
    "group":["IIB","IA","IIA","IB"]
  },
  {
    "name":"Advanced",
    "color":"rgb(255,0,0)",
    "group":["IIIIB","IV","IIIA"]
  },
  {
    "name":"Altri",
    "color":"rgb(0,0,255)",
    "group":["UNK","sconosciuto"]
  }
]
```

Il set di dati estratto dal database viene per motivi di compatibilità futura, convertito in formato XML e inviato alla pagina di elaborazione del grafico, che analizza i valori in

esso contenuti e genera i punti seguendo l'algoritmo genera i punti per disegnare il grafico.

```
<!--?xml version="1.0"?-->
<dataset>
  <data>
    <cc>20080154</cc>
    <data_diagnosi>Jan 21 2008 05:34PM</data_diagnosi>
    <stadio_clinico>IIIB</stadio_clinico>
    <stadio_patologico></stadio_patologico>
    <codice_diagnosi>162.8</codice_diagnosi>
    <codice_istologia>80123</codice_istologia>
    <os_stato>1</os_stato>
    <os_gg>691</os_gg>
    <abilitato>1</abilitato>
  </data>
  <data>
    ...
  </data>
  ...
</dataset>
```

L'algoritmo di Kaplan-Meier [KAP58] per la generazione delle curve di sopravvivenza, implementato in questo progetto, per motivi di spazio non può essere incluso all'interno di questo testo, viene però incluso il risultato in formato JSON:

```
{
  "punti":{
    "early": {
      "label": "Early = 100.0",
      "color":"rgb(0, 255, 0)",
      "data":[[[0,100],..., [44,100]]
    },
    "advanced": {
      "label": "Advanced = 100.0",
      "color":"rgb(255, 0, 0)",
      "data":[[[0,100],..., [44,100]]
    },
  },
}
```

```

    "altri": {
      "label": "Altri = 100.0",
      "color": "rgb(20,12,239)",
      "data": [[0,100], ..., [44,100]]
    }
  },
  "struttura": {
    "early": {
      "label": "Early",
      "color": "rgb(150, 150, 150)",
      "data": []
    },
    "advanced": {
      "label": "advanced",
      "color": "rgb(150, 150, 150)",
      "data": []
    },
    "altri": {
      "label": "altri",
      "color": "rgb(150, 150, 150)",
      "data": []
    }
  }
}

```


CAPITOLO 4.

CONCLUSIONI E SVILUPPI FUTURI

L'obiettivo della tesi è stato quello di realizzare un'applicazione web in grado di avvicinare e facilitare la consultazione di indicatori per il personale medico e di direzione della radioterapia dell'istituto di scientifico Romagnolo per lo studio e la cura dei tumori.

Per fare questo è stato necessario lavorare a stretto contatto con il personale medico della radioterapia e gli statistici dell'unità di biostatistica dell'IRST.

Questa interazione ha prodotto oltre a quanto visto nello svolgersi della tesi altre idee che vanno ad integrare il lavoro svolto e, viste le potenzialità dell'approccio scelto, sono in cantiere ulteriori sviluppi che ampliano lo spettro di ricerca al fine di comprendere meglio i risultati dell'attività svolta in reparto e lavorare per migliorarli.

Il sistema di autenticazione nato inizialmente per gestire il solo portale oggetto della tesi si è rivelato essere un progetto interessante e estendibile a tutti i microportali ad oggi presenti in intranet aziendale IRST.

Su di esso il servizio informatico dell'istituto ha intenzione di lavorare per creare una serie di contenuti interattivi a disposizione di tutti i dipendenti sia dall'interno dell'ospedale che dall'esterno.

Per quanto riguarda i possibili sviluppi futuri del portale, oltre all'integrazione delle estrazioni ed analisi effettuate estraendo i dati dal gestionale LOG80 all'interno del portale, si è già attivi per implementare una serie di indicatori statistici sulle curve di sopravvivenza.

Andando a calcolare questi valori in tempo reale, e quindi senza il bisogno di un interazione dell'unità di biostatistica il medico avrà informazioni più dettagliate per valutare le curve stesse.

Oltre a questo è stato chiesto di dare la possibilità di storicizzare le curve generate andando a salvare sia i parametri di estrazione che i dati in esse contenute in modo da poter ripetere con cadenza periodica le stesse estrazioni e confrontarle.

Per realizzare questa particolare richiesta si è pensato di oltre che implementare la suddetta funzione di salvataggio, di progettare e realizzare un'estensione in grado di permettere il "*merge*" di due o più estrazioni curve prodotte andando così a

BIBLIOGRAFIA

- [FLE81] Gilbert H. Fletcher, "History of Radiotherapy", 1981
- [ELE10a] Elekta Impac Software, "MOSAIQ User's Guide ver. 2.0", 2010
- [IRS12a] IRST, "R03P01 - Percorso paziente in Radioterapia", 2012
- [IRS12b] IRST, "R06P01 - Proposte studi clinici", 2012
- [NAN98] Nanni O., "Biostatistico.", Informatica in oncologia, pp. 82-89, 1998
- [POC83] Pocock SJ. "Clinical trials: a practical approach." Chichester, Wiley 1983.
- [ARM71] Armitage P. "Statistical methods in medical research." Blackwell Scientific Publications, 1971.
- [CUT58] Cutler SJ, Ederer F. "Maximum utilization of the life table method in analysing survival." J Chron Dis vol 8, pp. 699-712, 1958
- [KAP58] Kaplan EL, Meier P. "Nonparametric estimation for incomplete observation." J Am Stat Assoc, vol 53, pp. 457-481, 1958
- [LAW82] Lawless JS. "Statistical models and methods for life-time data." New York, NY, John Wiley and Sons, 1982.
- [PEA66] Pearson ES, Hartley HO. "Biometrika tables for statisticians." University Press (eds) Cambridge, 1966.
- [BER81] Bernstein L, Anderson J, Pike M. "Estimation of the proportional hazard in two-treatment-group clinical trials.", Biometrics vol 37, pp. 513-519, 1981.
- [MIE76] Miettinen OS. "Estimability and estimation in case-referent studies." Am J Epidemiol 1976 vol 103, pp. 226-235, 1976.
- [MAR95] Marubini E, Valsecchi MG. "Analysing survival data from clinical trials and observational studies." Chirchester: John Wiley & Sons, 1995.
- [LOG12] LOG80, "Gestione Sistema Qualità - Manuale Utente", 2012.
- [IRS12c] IRST, "R03P03 - Percorso ambulatoriale - rev. 1", 2012.
- [ELE10b] Elekta Impac Software, "MOSAIQ User's Guide ver. 2.0", 2010.
- [CLI11] K. S. Clifford Chao, CA. Perez, LW. Brady, "Radiation Oncology: Management Decisions", 2011.

- [WTS13] Web technology surveys, "Usage of server-side programming languages for websites", 2013
http://w3techs.com/technologies/overview/programming_language/all
- [PHP13] PHP: Hypertext Preprocessor, "PHP Manual", 2013
<http://www.php.net/manual/en/index.php>
- [JSO13] JSON, 2013
<http://www.json.org/>
- [W3C12] The World Wide Web Consortium, "Extensible Markup Language (XML)", 24/01/2012, <http://www.w3.org/XML/>
- [JQF13a] jQuery Foundation, "jQuery Framework Official Documentation", 2013, <http://docs.jquery.com>
- [JQF13b] jQuery Foundation, "jQuery Framework Official Documentation", 2013, <http://docs.jquery.com>
- [SPR13] SpryMedia, "DataTables Javascript API", 2013
<http://www.datatables.net/api>
- [LAU13] Laursen Ole, "Flot an Attractive JavaScript plotting for jQuery", 2013
<http://www.flotcharts.org/>
- [JUI13] "jQuery blockUI", 2013
<http://www.malsup.com/jquery/block/>
- [FUS13] "FusionCharts", 2013
<http://www.fusioncharts.com/>
- [COL11] "jQuery Plugin mColorPicker", 2011
<http://archive.plugins.jquery.com/project/mColorPicker>
- [EXP09] "Explorer Canvas", 2009, <https://code.google.com/p/explorercanvas/>