

UNIVERSITÀ DEGLI STUDI DI BOLOGNA

Facoltà Matematica fisica e scienze naturali

Corso di Laurea in 'Scienze di internet'

Banche del tempo

RELATORE

Prof. Danilo Montesi

TESI DI LAUREA DI

Danilo Centorbi

Matr. N. 282050

ANNO ACCADEMICO 2011-2012

Sessione I

Indice

1. Banca del tempo	4
1.1. Introduzione	4
1.2. Cosa è	5
1.3. Storia	5
1.4. Principi di fondo	6
1.5. Come funziona	7
1.6. Contesto giuridico	8
1.7. Stato dell' arte	8
1.8. Obiettivi del progetto	9
1.8.1. Soluzione	9
2. Social network	10
2.1. Web 2.0	10
2.1.1. Utente e contenuto	11
2.1.2. Socio-Semantic Web	12
2.2. Social network	12
2.3. Impatto Sociale	13
2.4. I Portali	13
2.4.1. Funzionamento di un portale	14
2.4.2. Funzionamento di una portlet	15
3. Analisi e progettazione	17
3.1. Requisiti generali	17
3.2. Requisiti del sistema	18
3.3. Requisiti dell' utente	19
3.4. La base di dati	21
3.4.1. Analisi dei dati	21
3.4.2. Schema relazionale	24
3.5. Requisiti funzionali	26
3.5.1. Analisi delle funzionalità	28

Indice

4. Ambiente e strumenti di lavoro	37
4.1. Liferay	37
4.1.1. I ruoli e i permessi	39
4.1.2. Creazione di una nuova community	40
4.1.3. Pannello di controllo ed installazione plugin e nuovi temi	41
4.2. MySQL	43
4.2.1. Integrazione con Liferay	44
4.2.2. Creazione tabelle	44
4.2.3. Campi personalizzati	45
4.2.4. Connessione al database	45
4.3. Vaadin	46
4.3.1. Integrazione con Liferay	47
4.3.2. Gestione dei temi	49
4.3.3. Add-ons	50
4.3.4. Gestione degli eventi con la libreria liferay-ipc	51
4.4. GoogleMap	52
5. Implementazione e integrazione con il portale	53
5.1. Architettura progetto	53
5.2. Struttura progetto	53
5.3. La base di dati	57
5.3.1. Integrazione con Liferay	57
5.3.2. Creazione tabelle mediante service builder	57
5.3.3. Campi personalizzati ed expando table	59
5.3.4. Schema relazionale	62
5.4. Vaadin e le portlet	63
5.4.1. Binding dei dati con Item e Container	64
5.4.2. Calendar Add-ons	69
A. Codice	70
A.1. Componente calendario	70
B. Regolamento Banca del tempo	72
C. Leggi in vigore	74

Elenco delle figure

2.4.1.Componenti della pagina di un portale	14
2.4.2.Diagramma di sequenza, ciclo di vita di una portlet	16
3.4.1.Schema relazionale base di dati	24
3.4.2.Schema relazionale base di dati rivisitato	25
3.5.1.Diagramma dei casi d' uso	27
3.5.2.Diagramma di sequenza, entità Utente	29
3.5.3.Diagramma di sequenza, entità Annuncio	31
3.5.4.Diagramma di sequenza, entità Messaggio	33
3.5.5.Diagramma di sequenza, entità Evento	35
3.5.6.Diagramma di sequenza, entità Feedback	36
4.1.1.Schema architettura ruoli in Liferay	39
4.1.2.Portlet , Particolare gestione community	41
4.1.3.Liferay, pannello di controllo	42
4.2.1.Entità coinvolte nella connessione tra MySQL e una web Application	43
4.3.1.Architettura vaadin	46
4.3.2.Portlet , liferay control panel	48
4.3.3.Architettura cartella temi di Vaadin	49
4.3.4.Architettura add-ons liferay-ipc	51
5.2.1.Diagramma di comunicazione , struttura applicazione	54
5.2.2.Diagramma delle classi , struttura progetto	56
5.3.1.Liferay, gestione campi personalizzati	59
5.3.2.Particolare Liferay, gestione permessi	60
5.3.3.Schema relazionale base di dati	62
5.4.1.Architettura vaadin	63
5.4.2.Architettura componenti Vaadin	64
5.4.3.Portlet , particolare inserimento annuncio	68
5.4.4.Portlet , particolare calendario gestione eventi	69

1. Banca del tempo

1.1. Introduzione

Con questo elaborato si vuole analizzare un progetto esistente da anni in Italia ed all'estero: la banca del tempo. Dopo aver conosciuto questa realtà, ho intervistato persone che promuovono attivamente e lavorano su questo progetto, nonché persone che vorrebbero entrare a farne parte. Nel corso di questa ricerca ho quindi iniziato a delineare una possibile implementazione di un sistema che potesse rispondere alle lacune strumentali a cui questa realtà deve far fronte ed ho deciso pertanto di creare un sistema di gestione per questo tipo di associazione che verrà in questo lavoro analizzata così come verranno descritte le fasi che hanno portato alla costruzione del sistema.

Il primo capitolo presenta l'associazione: 'la banca del tempo', spiegando le caratteristiche e i criteri di funzionamento; il secondo capitolo descrive, a grandi linee, il concetto di rete, nello specifico il Web 2.0 e come esso ha modificato i criteri e le modalità di comunicazione tra le persone. Verrà nel corso del capitolo, delineato il portale e il funzionamento di esso, perchè il progetto sarà sviluppato su questa piattaforma. Il terzo capitolo riguarda l'analisi del problema affrontato, per chiarire la struttura dell'applicazione sviluppata. Nel quarto capitolo si entrerà nello specifico della messa in opera di un portale, descrivendo i passi per la creazione di pagine web, delle loro componenti, le portlet, e il modo con cui esse comunicano con la base di dati. Il quinto capitolo mostrerà i metodi di integrazione e sviluppo dei componenti di cui sarà composto il sistema, mentre il sesto ed ultimo capitolo verteranno sulla descrizione dell'implementazione del progetto.

1. Banca del tempo

1.2. Cosa è

La Banca del Tempo è un istituto di credito nel quale si deposita denaro e non si riscuotono interessi, ma la disponibilità a scambiare prestazioni con gli altri aderenti, utilizzando il tempo come unità di misura degli scambi, secondo il principio che un'ora di un tipo di servizio è equivalente all'ora di qualsiasi altra tipologia di servizio scambiato.

Ad ognuno degli aderenti viene intestato un regolare conto correntetempo e viene consegnato un libretto di assegni tempo: unico obbligo è il pareggio. La Banca del Tempo è un'innovazione sociale che attiva una rete di solidarietà fondata sullo scambio alla pari di prestazioni capaci di soddisfare bisogni legati alla vita quotidiana. Un modo per riorganizzare la rete di reciproco aiuto, tipica dei rapporti di buon vicinato sviluppando l'abitudine alle pratiche di reciprocità solidale e di mutuo aiuto. Quello ecosolidale ed ecocompatibile è un modello alternativo che tende a valorizzare le persone con tutte le loro differenze, le varie culture e le diverse etnie, che si realizza attraverso transazioni regolate in termini di tempo; un'occasione in cui le persone hanno la possibilità di mettere a disposizione della collettività le proprie capacità e i propri saperi e, allo stesso tempo, possono soddisfare i propri bisogni e desideri.

Se pure le BdT non usano denaro, contribuiscono a pieno titolo alle dinamiche economiche, creando un tipo di economia alternativa che usa come moneta complementare il tempo.

1.3. Storia

Il termine "Banca del Tempo" viene usato per la prima volta a Parma agli inizi degli anni '90, inizialmente era un'associazione aperta alle sole donne, formatasi per venire incontro alle necessità di quelle persone, anziani e pensionati, che, sentendosi esclusi dalla vita sociale, avrebbero potuto sfruttare il tempo a loro disposizione in maniera utile. Sarà poi la sperimentazione effettuata da un gruppo di donne di Santarcangelo di Romagna a far conoscere a livello nazionale ed internazionale il progetto. Già da queste esperienze si è compreso che il valore reale di un'organizzazione di tale genere dipende direttamente dal numero di soci che contribuiscono attivamente al progetto, quindi nel 1995, attraverso una serie di incontri divulgativi, l'idea di scambio di tempo incontra il favore di numerosi gruppi (associazioni costituite e gruppi informali) che nel breve periodo danno vita a Banche del Tempo; alla fine del 1995 un organismo nato da un accordo con l'Università di Siena e la CGIL (Sindacato) "Il Cittadino ritrovato" costituisce l'osservatorio nazionale che ha sede in Roma, presso la CGIL nazionale. Tempomat ha il compito di diffondere la conoscenza delle BdT, di raccogliere i dati sulle esperienze in atto, di sostenere con materiale l'apertura di BdT, di organizzare convegni, di attivare laboratori di formazione. Le banche del tempo così

1. Banca del tempo

nascono e crescono, ma il fenomeno si sviluppa soprattutto nel centro e nel nord d'Italia; in un anno vengono censite sessantaquattro banche, ma, alla fine del 2002, anno in cui l'osservatorio in CGIL chiude, si contano circa 250 banche. La CGIL nazionale, in Roma, Corso Italia n. 25 ha cessato la sua attività con la fine di Ottobre 2002, per chiusura del centro medesimo. Il logo è stato attribuito alla Banca del Tempo: "Le città invisibili" di Guspini, per essere utilizzato sul sito "www.tempomat.it" di proprietà della stessa banca ed è tuttora in funzione. Con la chiusura dell'osservatorio nazionale si perde tutto un lavoro di raccordo, di sostegno alle banche che si organizzano nei propri territori, ma stentano a relazionarsi, ad incontrarsi, fatti salvi alcuni convegni organizzati spesso dagli enti locali. Si perde il senso della relazione, della condivisione e del configurarsi dell'esperienza.

Sul bisogno di riprendere quel lavoro e quella relazione nasce l'Associazione Nazionale nel 2007, che permette alle banche di prendere vigore e di attecchire anche nel sud d'Italia e si avvia pertanto quel processo di rete, che mette in relazione banche del tempo, cittadini, associazioni, enti locali. Naturalmente il processo è iniziato da pochi anni e lo stesso percorso formativo si sta costruendo con fatica e studio, perché se la diversità delle banche è ricchezza, la costruzione di una linea comune è fondamentale e va tessuta nel tempo.

1.4. Principi di fondo

La Banca del Tempo si basa sullo scambio, cioè si dà per ricevere, si chiede tempo per restituirlo; il proprio conto corrente, in linea di massima, dovrebbe tendere ad avere saldo zero dopo un certo periodo di collaborazioni. Non si tratta di volontariato, ove i volontari offrono tempo per loro attività e non ricevono niente, se non la mera gratificazione per l'azione svolta. Si parla di reciprocità indiretta, ogni scambio infatti innesca un meccanismo di debiti e di crediti in tempo nei confronti della Banca, non del singolo interessato. Il credito registrato nel conto corrente di chi ha offerto la prestazione non dovrà necessariamente essere utilizzato nei confronti della stessa persona, ma nei confronti di qualsiasi altro utente della banca. Allo stesso modo si potrà rientrare dal debito offrendo prestazioni ad altri associati che richiedono il servizio. Il tempo è l'unità di misura: il valore della prestazione è determinata dal tempo impiegato nello scambio non dal tipo di servizio offerto, a differenza di altre realtà, come la Bdt inglese (LETS) in cui il valore delle prestazioni e dei beni scambiati è strettamente correlato a quelli di mercato. Nella Banca del Tempo il solo denaro ammesso è quello versato a titolo di rimborso spesa. Tutti gli aderenti sono uguali tra loro. Un'ora è sempre tale indipendentemente dall'età, dalla scolarità, dal ceto sociale di chi l'ha scambiata ed indipendentemente dal tipo di prestazione domandata od offerta. Il territorio di riferimento della Banca del Tempo è solitamente limitato: un quartiere in una grande città, un piccolo comune, una scuola, un circolo ricreativo. La piccola dimensione facilita la socialità ed elimina le difficoltà di spostamento, ma, con il tempo, tra le esperienze più

1. Banca del tempo

mature, si sono attivati scambi anche tra le diverse Banche, ed è in questa direzione che le associazioni italiane si prodigano nello sviluppo di questa forma di economia alternativa.

1.5. Come funziona

Chi si iscrive alla Banca del Tempo definisce le prestazioni che intende offrire, indicando anche quelle delle quali pensa di avere bisogno; sebbene la lista è puramente indicativa tuttavia rappresenta un punto di partenza per dar vita a richieste ed offerte. Definire bisogni e disponibilità permette di evidenziare fin da subito che lo scambio, di qualsiasi tipo, sta alla base dell'esperienza sociale a cui fa riferimento questo esperimento. Ad ogni iscritto, che entrando si assume definite responsabilità leggendo e sottoscrivendo il regolamento interno dell'associazione e, talvolta, in base all'associazione di cui entra a far parte, stipula un contratto assicurativo, viene intestato un conto corrente in tempo nel quale verranno registrate le ore addebitate ed accreditate, con l'impegno di movimentare il conto sia in un senso che nell'altro. Principali strumenti operativi quindi sono: il libretto degli assegni, la lista delle offerte e delle richieste, il regolamento interno definito ed aggiornato dall'assemblea degli associati, la lista dei partecipanti con i riferimenti per chiamare direttamente (in alcune BdT tale lista rimane depositata presso lo sportello e gli scambi sono necessariamente da questo mediati). La richiesta della prestazione potrà essere quindi effettuata direttamente all'offerente o allo sportello che gestisce la banca. Non vi è alcun obbligo a soddisfare la prestazione (se in quelle ore o in quel giorno non è possibile, si può rifiutare lo scambio). Chi riceve la prestazione emette un assegno in tempo che verrà recapitato presso la segreteria della BdT da chi l'ha ricevuto. L'assegno permette la contabilizzazione dello scambio, a chi ha effettuato la prestazione verrà accreditato quindi il tempo dovuto, viceversa per chi ne ha usufruito. Il credito, o il debito, potrà essere speso o restituito scambiando con altri membri del gruppo, perché matura nei confronti della Banca del Tempo e non del singolo aderente. Unico impegno è quello di tendere al pareggio: non sono positive né le posizioni troppo a credito né quelle troppo a debito, perché indicano chiaramente che non vi è scambio! Tutte le prestazioni sono valutate in tempo, anche nelle attività di segreteria, ad esempio, non circola denaro, se non quello a copertura delle spese vive, per effettuare lo scambio.

Le prestazioni scambiate dipendono dalle abilità, dalle disponibilità e dalla fantasia degli aderenti. Nella pratica poi, bisogni estemporanei non indicati nelle disponibilità offerte, o nelle richieste abituali, possono essere presentati al gruppo trovando così soluzione. A differenza di quanto avviene sul mercato, ove le offerte si selezionano in base alle richieste di mercato ed in assenza (o limitata richiesta) di "consumatori", non trovano una risposta economica, nella Banca del Tempo quello che interessa è la relazione con gli altri. Alcune Banche del Tempo hanno introdotto il prestito di beni, altre lo scambio di beni. A tutt'oggi comunque appare prevalente lo scambio di tempo per prestazioni (98%).

1.6. Contesto giuridico

Le Banche del Tempo realizzano una concreta attuazione dei principi di eguaglianza e di solidarietà tra i cittadini riconosciuti negli art. 2 e 3 della Costituzione, poichè contribuiscono a realizzare ed a diffondere nuove forme di aiuto reciproco, utilizzando il tempo come unità di misura delle attività scambiate. Giuridicamente sono "associazioni di promozione sociale" e non ONLUS; come spiega la legge 383 del 7 Dicembre del 2000, le associazioni dei datori di lavoro, le associazioni professionali e di categoria e tutte le associazioni che hanno come finalità la tutela esclusiva di interessi economici degli associati, escluse da quest' ultime sono i partiti politici.

Attualmente in Italia non vige una precisa e definita legislatura riguardo le Banche del tempo, ma ci si può rifare alle leggi sulle associazioni di promozione sociale per definire e chiarire quali sono le normative che regolano questa associazione. In appendice C l' attuale regolamentazione riguardo le banche del tempo. Dagli articoli si evince che lo stato promuove questa tipologia di associazione ma, all' atto pratico, questi gruppi mancano di aiuti ed incentivi concreti per la promozione e lo sviluppo di queste associazioni.

Dagli articoli si evince che lo stato promuove questa tipologia di associazione ma, all' atto pratico, questi gruppi mancano di aiuti ed incentivi concreti per la promozione e lo sviluppo di queste associazioni.

1.7. Stato dell' arte

Nel territorio italiano sono molte le banche del tempo attive, suddivise per regione, anche perchè ogni regione attribuisce diverso peso e incentivo alla costituzione di questo tipo di associazione, risulta subito evidente, da una prima analisi del fenomeno, che nel nord Italia questo progetto riscontra più successo. Come punto di riferimento si fa fede alla realtà presente nel territorio Emilia Romagna, dove sono presenti in totale 42 BdT. Queste associazioni sono gestite da volontari in maniera democratica e in assenza di gerarchie definite, che purtroppo, soprattutto nell' ultimo periodo, non riscontrano aiuti da parte della pubblica amministrazione.

Negli anni precedenti la regione Emilia Romagna ha reso disponibile un software per la gestione degli utenti e degli annunci utile alla gestione dei soci delle BdT e dei crediti a loro riferiti, ma oggi questo software è divenuto obsoleto e la regione non offre supporto alle richieste di aggiornamento dello stesso, anche per quanto riguarda il semplice aggiornamento delle categorie delle offerte o richieste inserite dai soci.

1. Banca del tempo

Il territorio di riferimento della Banca del Tempo è solitamente limitato. La piccola dimensione, difatti, facilita la socialità ed elimina le difficoltà di spostamento, ma con il tempo, tra le esperienze più mature, si sono attivati scambi anche tra le diverse Banche presenti in territori per lo più adiacenti tra di essi.

1.8. Obiettivi del progetto

Lo sviluppo di questo progetto nasce dalla necessità di creare una piattaforma che possa divenire un punto di riferimento per i soci delle varie BdT, in grado di connettere i soci esistenti e dare loro un valido strumento di gestione degli scambi, offrendo nel contempo uno strumento utile a favorire lo sviluppo della rete sociale dando la possibilità di conoscere nuovi utenti. Con questo progetto si vogliono abbattere le difficoltà che questo tipo di associazioni incontrano, che risiedono principalmente nell' utilizzo di strumenti di gestione obsoleti e non funzionali, nonché localizzati ad aree geografiche relativamente piccole. Da qui l' idea di creare un' applicazione accessibile mediante un sito che possa divenire un punto di riferimento per i soci che, essendo già a conoscenza del progetto vogliono aggregare gli sforzi e favorire lo sviluppo di questa tipologia di economia.

1.8.1. Soluzione

Per offrire ai soci della BdT un modo per conoscere persone e rimanere in contatto con loro, la creazione di un Social Network Sites è la migliore soluzione per offrire all' utente una via, oramai di uso comune, per rimanere in contatto con altri utenti che condividono gli stessi interessi, per estendere la propria rete di contatti e per creare una comunità che, nello specifico, si propone attivamente garantendo lo scambio e la solidarietà come principio di fondo per una convivenza fondata sull' aiuto reciproco; fermo restando il fatto che ad un'azione ne corrisponde un' altra.

Il progetto è volto alla creazione di una rete sociale digitale che rispecchi una rete sociale reale. Nel prossimo capitolo verranno descritte le tecnologie attualmente disponibili per la creazione di reti sociali digitali.

2. Social network

In questo capitolo si vogliono, brevemente, introdurre gli strumenti e le tecnologie web di uso comune, evidenziando l' impatto che l' utilizzo di tali tecnologie ha avuto tanto nel navigatore, quanto nei contenuti scambiati da esso all' interno della rete. Definendo il concetto di web 2.0 si passerà ad analizzare come tali tecnologie hanno modificato le modalità, nonché la quantità e la qualità delle informazioni scambiate.

Nel corso del capitolo si chiariranno i concetti di social network e gli strumenti utilizzati solitamente dagli sviluppatori per la creazione di questo tipo di siti web: i portali.

2.1. Web 2.0

Web 2.0: Stuff that allows users to create content or share content with a pastel palette, big fonts and rounded corners.

Eric G. Myer

Il principale motivo per cui gli sviluppatori hanno adottato e contribuiscono a migliorare continuamente queste tipologie di piattaforme è quello di sfruttare e amplificare gli effetti di rete che caratterizzano il web. Più persone usano una determinata tecnologia, maggiori sono i benefici che ne traggono i fruitori stessi. Attualmente, grazie anche all' innovazione ed agli incentivi offerti dai produttori di strumenti tecnologici, in collaborazione con le compagnie di telefonia mobile in Italia, come all' estero, gli utenti diventano sempre più numerosi e continuamente connessi alla rete, aumentando esponenzialmente la quantità di informazioni scambiate.

Il Web 2.0 è dunque un luogo di collaborazione che consente un elevato livello di interazione tra sito e utente e che permette elevata partecipazione e collaborazione nello scambio d'informazioni all'interno della rete internet; in generale quindi, esso non è solamente composto di contenuti strutturati in un preciso modo, ma di funzionalità disponibili ed accessibili da diversi device, e riutilizzabili su diverse interfacce embedded.

2. Social network

2.1.1. Utente e contenuto

Il fulcro della rete non è più il contenuto, come nel web statico; ma chi pubblica le informazioni, chi le legge e chi le commenta; chi cioè condivide le proprie idee. Con questo tipo di approccio, il contenuto non rimane uno strumento di comunicazione tra 2 attori (il mittente ed il destinatario), ma viene arricchito da tutte le persone che hanno qualcosa da esprimere agli altri.

La svolta digitale e la crescita esponenziale delle possibilità di accedere e condividere i contenuti cambiano radicalmente il modo di interagire con i dati, le informazioni e la conoscenza.

si nota difatti:

- un'accelerazione di diffusione delle informazioni.
- contenuti di minori dimensioni ma più utili(dettagliati) e riutilizzabili.
- una popolazione più ampia di persone che producono e condividono informazioni(piccoli/anziani).
- crescente specializzazione degli argomenti, interessi secondo nicchie sociali.
- crescita massiccia di spazio per l' azione, e la libertà di espressione a disposizione di ogni singolo individuo.

Vengono così a formarsi innumerevoli gruppi di utenti che condividono le stesse idee, gli stessi interessi, le stesse passioni; la rete diventa un punto di incontro in cui è possibile conoscere individui e aggregarsi per soddisfare il proprio bisogno di conoscenza. I gruppi che vengono a formarsi, le comunità, che si costituiscono all' interno della rete, divengono autogestite, autoregolamentate e autocostruite mediante gli attori stessi che le compongono. Le strutture, nonché le applicazioni e gli strumenti implementati, per soddisfare tali esigenze, hanno, quindi, bisogno del continuo apporto di informazioni da parte dell' utente, sia per motivi cognitivi che cooperativi, ma al momento, il Web ha una conformazione confusionaria, dove sempre più gente condivide pubblicamente su un scala globale. L' evoluzione che si prospetta più adeguata a questo cambiamento risiede nell' implementazione tramite le linee guida del "Socio-Semantic Web".

2. Social network

2.1.2. Socio-Semantic Web

Il "Socio-Semantic Web" (S2W) aggrega le tecnologie e le metodologie proprie delle recenti piattaforme web allo scopo di costruire artefatti simbolici e organizzativi, comprendenti mappe e altri strumenti semantici, come i tag o i sistemi di bookmarking, volti a modificare la visibilità, l'organizzazione e la rappresentazione della conoscenza e delle azioni all'interno di una comunità. Il "Socio-Semantic Web" presenta una visione delle informazioni sul web con una strutturazione indipendente, formalizzata in uno specifico linguaggio, al fine di facilitare la ricerca, la condivisione e l'integrazione delle informazioni, consentendo una più sofisticata e dettagliata analisi dei dati mediante agenti specializzati.

In questo modo le persone accedono e creano facilmente i contenuti mentre le macchine analizzano e catalogano in maniera più fluida e dettagliata. Il risultato potrebbe essere un web strutturato in frammenti di informazione e canali, resi riutilizzabili, remixabili, combinati e annotati con più riferimenti incrociati, che consentano a tutti di produrre informazioni rilevanti solo una volta, ma accessibili e condivisibili da contesti e con individui e gruppi diversi. Rispetto al Web 2.0 avrà un orientamento più locale, di nicchia; in contrasto con la trasmissione tutto-a-tutti, questa 'metodologia' consentirà di avere una struttura ancora più user-centered.

Ogni utente può, difatti, accedendo ad una singola risorsa, accedere ai contenuti della rete intera, mediante un sistema di bookmarking 'intelligente' che valorizza l'utente, i propri interessi, le proprie necessità e le proprie scelte. Questo tipo di approccio è già presente in quelle aziende che necessitano di suscitare collettivamente, in modo continuo, una parte fondamentale della conoscenza, in particolare della struttura "locale" semantica di specifici oggetti di business e di lavoro collettivo, ma potrebbe rendersi un'ottima soluzione per la riorganizzazione della rete internet.

2.2. Social network

La nascita dei social network indica un cambiamento organizzativo. Le comunità on-line iniziali come usenet e forum di discussione erano strutturate per argomenti o per gerarchie di argomenti gli SNS invece sono strutturati come reti personali o egocentriche con l'individuo al centro della propria comunità.

Leggendo la definizione data dagli studiosi Boyd-Ellison si possono definire social network sites quei servizi web che permettono di gestire e rinsaldare on-line amicizie preesistenti o di estendere la propria rete di contatti costituendo un gruppo. Le comunità virtuali sono quei gruppi di persone che interagiscono mediante mezzi di comunicazione come newsletter,

2. Social network

mailing list, forum, instant messaging, wiki, e social network, attraverso cioè le piattaforme e i media in costante aumento messi a disposizione da internet.

Questa tipologia di sito/applicazione è la concreta rappresentazione del concetto di Web 2.0, che come prima discusso, permette all'utente di creare attorno a sé una rete di amicizie e di informazioni strettamente personalizzata. Gli sviluppatori delle più famose ed utilizzate piattaforme sociali tendono difatti ad includere nelle proprie applicazioni quanti più servizi utili agli utenti per gestire la propria rete di contatti e rendere questi ultimi funzionali alla propria vita.

2.3. Impatto Sociale

Con Web 1.0 solo gli utenti esperti erano in grado di creare e gestire un sito, in quanto era necessaria una buona conoscenza sia di HTML sia di linguaggi informatici e questo ne limitava la diffusione. I blog prima di tutti hanno semplificato la creazione di pagine interattive: chiunque può pubblicare sul proprio sito contenuti dinamici di grande effetto senza necessariamente possedere particolari conoscenze tecniche; questo ha dato la possibilità a tutti di condividere informazioni con altri utenti. Il progressivo passaggio dal web 1.0 al 2.0 ha quindi conseguenze positive anche dal punto di vista sociale. La semplicità di utilizzo, e il poter pubblicare anche senza conoscere linguaggi complicati, annessi alla possibilità di separare le informazioni dalla forma, hanno decretato il successo delle piattaforme gratuite che contano un numero esorbitante di iscritti e di partecipanti attivi.

2.4. I Portali

Il portale è un servizio che opera da mediatore di informazione a favore degli utenti della rete, permettendo a questi di raggiungere, tramite un particolare punto di ingresso nella rete, una grande quantità di risorse esistenti. Un portale è quindi un aggregatore di informazione, un servizio di navigazione sul web che facilita il lavoro di ricerca: nasce infatti come un motore di ricerca, e si arricchisce in seguito di servizi, informativi e non, allo scopo di proporsi come accesso preferenziale e guida per la navigazione via Internet.

Oggi i portali sono spesso utilizzati per la costituzione di “applicazioni sociali”, siti Internet che offrono all'utente, tra gli altri, numerosi servizi di comunicazione e scambio di informazioni con altri utenti.

2.4.1. Funzionamento di un portale

I portali per rendere i servizi prima esposti si compongono di particolari componenti software: le portlet. Una portlet è un modulo web realizzato in tecnologia Java, riusabile all'interno di un portale utile, a generare contenuti dinamici e altamente personalizzabili. Diverse portlet, integrate insieme, concorrono a creare una pagina del portale, caratteristica fondamentale è la possibilità da parte dell'utente di scegliere quali portlet compongono la pagina per offrire un'esperienza di navigazione altamente personalizzabile.

All'utente il compito di scegliere i componenti che formeranno la pagina web, scegliendo la posizione e talvolta l'aspetto grafico, il portlet container assolve, invece, ai compiti infrastrutturali necessari al corretto funzionamento di questi elementi di interfaccia: gestire l'impaginazione, immagazzinando le preferenze e le configurazioni di ogni portlet e gestire il loro ciclo di vita, operando da mediatore. Il portlet container riceve difatti le richieste dal portale e le indirizza alle portlet che le gestiscono. Con le portlet è possibile comporre in un'unica pagina funzionalità distinte che operano su sorgenti dati diverse. Ciò consente di aggregare, in una interfaccia omogenea, dati e applicazioni eterogenee, senza costringere l'utente a viaggiare tra applicativi diversi per eseguire le funzionalità di cui ha bisogno, rendendo la navigazione più fluida e funzionale.

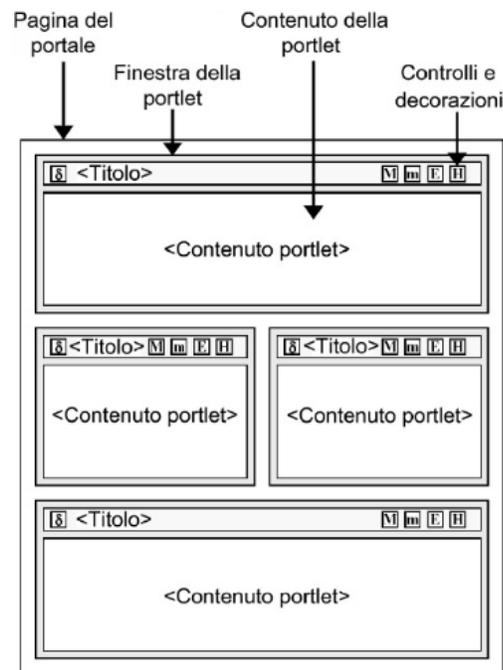


Figura 2.4.1.: Componenti della pagina di un portale

2.4.2. Funzionamento di una portlet

Una portlet è una componente di una pagina, è quindi responsabile solo di una porzione di pagina web, il ciclo di gestione della richiesta a cui deve far fronte il portlet container è dunque piuttosto articolato. Si distinguono diverse fasi nella gestione della richiesta.

- Render Phase

È la fase che va in esecuzione ogni volta che la portlet viene visualizzata nella pagina web.

- Action Phase

È la fase durante la quale la portlet esegue delle azioni che provocano un cambiamento nel suo stato, che poi si riflette nella fase di render.

- Event Phase

È la fase mediante la quale la portlet gestisce gli eventi per i quali è in ascolto.

- Resource Serving Phase

È la fase in cui una portlet serve all'utente una particolare risorsa, quale ad esempio un file.

Una portlet ha diverse fasi nella gestione di una richiesta, ed ha anche un comportamento diverso a seconda del tipo di richiesta che viene effettuata. A una richiesta di azione ci sarà l'esecuzione della action phase seguita dalla render phase, a una richiesta di visualizzazione del proprio contenuto, seguirà soltanto la fase di render. Essendo una pagina composta da diverse portlet, ognuna responsabile di un frammento di pagina, ad una richiesta da parte dell'utente di un'azione su una singola portlet risponderà solo la portlet in questione, tutte le altre semplicemente si renderanno fronte di questa richiesta. Per cui la portlet sulla quale è stata fatta l'azione attraverserà le fasi di action e di render, le altre solo quella di render.

Il diagramma in figura 2.4.2 mostra il comportamento appena descritto.

Il portlet container dunque gestisce le richieste e il ciclo di vita delle portlet, le classi java che vogliono inviare le richieste devono implementare delle interfacce specifiche. Ogni portlet deve implementare l'interfaccia `javax.portlet.Portlet` e può, opzionalmente, implementare le interfacce `javax.portlet.EventPortlet` e `javax.portlet.ResourceServingPortlet`.

L'interfaccia `javax.portlet.Portlet` definisce infatti i metodi principali del ciclo di vita di una portlet:

- `init()`

2. Social network

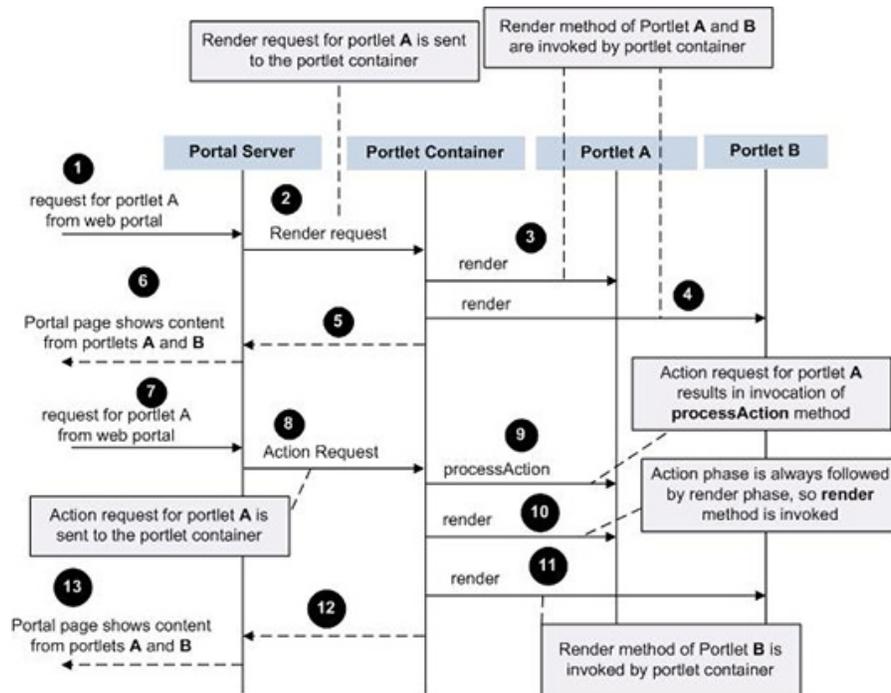


Figura 2.4.2.: Diagramma di sequenza, ciclo di vita di una portlet

È il metodo invocato dal container dopo che la portlet viene caricata e istanziata e consente di effettuare operazioni di inizializzazione.

- render()

È il metodo responsabile della generazione del contenuto della porzione di pagina gestita dalla portlet. Va in esecuzione a seguito di una richiesta di tipo render.

- processAction()

È il metodo che va in esecuzione a seguito di una azione dell'utente sull'interfaccia, quindi a seguito di una richiesta di tipo action.

- destroy()

È il metodo invocato dal container quando la portlet viene rimossa dalla memoria e può essere usato per operazioni di deallocazione di risorse.

Avendo brevemente descritto il funzionamento dei portali e dei loro componenti, le portlet, è adesso possibile analizzare i requisiti dell'applicazione che si vuole sviluppare.

3. Analisi e progettazione

Il progetto ha come base operativa la città di Bologna, ma mira a divenire un punto di riferimento per tutte le persone che vogliono contribuire attivamente a questa iniziativa. A differenza di svariati paesi europei, in Italia questo tipo di associazione è molto localizzata, la speranza è quella di offrire uno strumento utile per l'espansione della Banca del Tempo, con la speranza che sempre più persone possano rendersi conto che è possibile trovare ciò di cui abbiamo bisogno dai nostri 'vicini'.

Questa prima fase di progettazione è volta all'individuazione dei requisiti, ponendo attenzione a quelle che sono le richieste dell'utente e del sistema, attori principali del sistema implementato. Da una generale individuazione del modello di dominio si andranno, nel corso del capitolo, a delineare le caratteristiche statiche e dinamiche della struttura, della base di dati e delle funzionalità di cui il sistema è composto.

3.1. Requisiti generali

La BdT si fonda essenzialmente su due entità: i soci, con il tempo da loro condiviso e la lista delle richieste e delle offerte da loro condivise. Prendendo come punti di riferimento le BdT esistenti in Italia ed all'estero possiamo definire il modello della realtà da modellare, in generale, per poi definire nello specifico, tutte le funzionalità da implementare.

Ogni socio deve avere la possibilità di consultare la lista di tutti gli utenti presenti nella comunità e la lista degli annunci da loro inseriti. Al momento della creazione dell'account ogni membro della comunità arricchisce il proprio profilo con informazioni personali e la propria posizione geografica, queste informazioni dovranno essere sempre aggiornabili e permetteranno di dare un'istantanea dell'utente all'interno del gruppo.

Quando una persona, consultando la pagina relativa ad un utente, o la lista degli annunci, trova un'offerta o una richiesta interessante, si iscrive ad essa, selezionando la data in cui intende effettuare lo scambio. L'utente che ha inserito l'annuncio viene, dunque, notificato dell'interesse in questione ed ha la possibilità di confermare la data inserita, o modificarla, in base ai propri impegni ed esigenze; se lo scambio fosse inattuabile, potrà essere eliminato.

3. *Analisi e progettazione*

Nella propria pagina personale ogni utente ha quindi un calendario che tiene nota di tutti gli impegni presi con gli altri utenti.

Se tra gli annunci inseriti non compare nessuna offerta utile a soddisfare i bisogni di un utente, quest'ultimo può inserire una richiesta ed attendere che qualche utente sia in grado e disposto a soddisfarla. Per ovviare al problema che si pone al momento di attendere la risposta ad una richiesta specifica e per semplificare il sistema degli scambi, si è scelto di far coincidere le categorie selezionabili degli annunci, con gli interessi e le abilità di un utente. Una persona che, quindi, cerca qualcuno in grado, ad esempio, di organizzare un evento, potrà consultare le pagine degli utenti della propria zona e inviare un messaggio agli utenti che hanno interesse nella categoria 'organizzazione'.

Se qualche utente, al momento della creazione del proprio account, o quando inserisce un annuncio, si rende conto che tra le categorie disponibili non compare una specifica categoria, potrà inviare un messaggio all'amministratore chiedendo di aggiungere la suddetta categoria alla lista di quelle disponibili.

Dopo avere effettuato uno scambio, l'utente, che ha ricevuto il servizio specifico ha la possibilità di lasciare un voto e un breve commento riguardo lo scambio in questione. Tutti i commenti verranno raccolti e saranno così consultabili nelle pagine di ogni utente.

3.2. **Requisiti del sistema**

Il sistema sarà utilizzabile da qualsiasi dispositivo che disponga di una connessione di rete, non è richiesta un'elevata banda di connessione, in quanto si prevede per gran parte l'utilizzo di linguaggio Ajax per lo sviluppo degli applicativi; alle richieste di inserimento dati e di consultazione delle informazioni, il sistema elaborerà per il browser richiedente soltanto i dati necessari all'elaborazione della richiesta stessa, ed aggiornerà quindi soltanto i componenti interessati.

L'interfaccia utente deve essere usabile, di semplice apprendimento ed utilizzo, per permettere a differenti fasce di utenti l'utilizzo senza problemi dell'applicativo. Essendo accessibile da tablet e smartphone, gli agenti che compongono l'applicativo avranno una grafica, nonché un'interfaccia minimale. Il sistema implementato dovrà essere modulare e riusabile, per permettere e non impedire l'evoluzione e la creazione di servizi aggiuntivi agli utenti. Altra caratteristica da tenere in considerazione è la personalizzazione della propria area, sarà necessaria per una maggiore utilità del sistema, ad esempio modificare la posizione e l'aspetto grafico (colori, caratteri) dei componenti della pagina o aggiungere e rimuovere alcuni servizi messi a disposizione da esso.

3. *Analisi e progettazione*

La struttura di navigazione deve incoraggiare la ricerca e l'esplorazione, deve fornire continui stimoli all'utente oltre a guidarlo in modo chiaro alla navigazione, facendo sì che possa giungere facilmente alle informazioni ricercate. I criteri di strutturazione devono essere pensati in modo che risultino chiari al pubblico, che dovrà essere aiutato e incoraggiato nell'esplorazione e nel facile recupero delle informazioni che riguardano le persone che compongono la comunità e gli annunci presenti. Anche la gestione degli annunci, degli appuntamenti presi e dei messaggi deve essere immediata, ma questi aspetti si discuteranno più avanti; in questa sede definiamo che le componenti (le portlet di cui si è discusso precedentemente) del sistema, saranno lo strumento attraverso il quale l'utente potrà interagire con il sistema; questa strutturazione renderà il sistema flessibile e dinamico, sarà possibile, senza variazioni strutturali importanti, modificare o aggiungere nuove funzionalità, all'occorrenza, e riuscirà, al meglio, a soddisfare i requisiti richiesti.

3.3. **Requisiti dell'utente**

Il socio della BdT, da ora in avanti utente, è un membro attivo della comunità di cui fa parte. A lui è permesso accedere alle informazioni riguardanti gli altri utenti, può comunicare con essi per scopi inerenti al progetto BdT o semplicemente per instaurare, inizialmente, un rapporto di amicizia. Per rendere possibile lo scambio di informazioni utili al progetto, al momento della creazione del proprio account, la persona interessata compila un form con i propri dati personali e del profilo, che dovranno essere aggiornabili in qualunque momento, un'immagine personale aiuterà l'utente ad avere memoria di un contatto già visualizzato precedentemente; al momento dell'accettazione delle norme di condotta della comunità, a cui l'utente dovrà far fede rispettandole nel rispetto degli altri membri della comunità, esso avrà diritto ad uno spazio pubblico e ad uno spazio riservato, quindi avrà la possibilità di navigare tra le pagine riservate ai membri della comunità. Lo spazio riservato sarà visibile solo all'utente loggato; attraverso questo spazio sarà possibile avere un'istantanea delle proprie attività e sarà possibile inserire nuovi annunci o gestire le richieste inviate dagli altri utenti. Le pagine pubbliche serviranno invece per rendere disponibile il proprio profilo agli altri utenti, offrendo informazioni utili all'instaurazione di nuove amicizie e di nuovi contatti.

In dettaglio le tipologie di utenti:

- visitatore, chi seguendo un link o un consiglio visita per la prima volta le pagine della Banca del tempo.
- utente loggato, socio della Banca del tempo che ha creato l'account e usufruisce dei servizi offerti.

3. *Analisi e progettazione*

- amministratore, socio che gestisce i permessi e alcuni contenuti relativi ai soci della Banca del Tempo.

Per far fronte alla necessità di poter visualizzare dettagliatamente le informazioni dei soci della Bdt, e gestire le inserzioni, nonché i messaggi e gli eventi che un socio inserisce per programmare una transazione, si è scelto di sfruttare la strutturazione del portale che mette a disposizione di ogni utente loggato due tipologie di pagine, la pagina pubblica e la pagina privata; di seguito verranno delineate le funzionalità che questo tipo di impostazione mette a disposizione dell'utente.

pagina pubblica

La pagina visibile a tutti i membri della community, durante la navigazione tra le pagine del sito l'utente avrà la possibilità di essere indirizzato verso le pagine pubbliche dell'utente di cui chiede maggiori informazioni, nelle suddette pagine saranno visibili le informazioni del profilo dell'utente in questione:

- interessi.
- annunci inseriti, sia richieste che offerte.
- feedback ricevuti ed inviati.
- voti dagli altri utenti per le transazioni eseguite e per il profilo in genere.
- commenti dagli altri utenti.

pagina riservata

La pagina personale di ogni utente, attraverso questa pagina, visibile soltanto dall'utente loggato, è possibile gestire:

- messaggi ricevuti ed inviati.
- calendario con appuntamenti ricevuti, quindi da confermare, e appuntamenti inseriti, per cui si attende conferma.
- gestione, quindi creazione, modifica ed eliminazione annunci.
- inserimento feedback per transazioni eseguite.
- visualizzazione conto.

3.4. La base di dati

Nelle fasi iniziali del progetto ho intervistato diverse persone per avere un quadro generale riguardo la struttura che questo tipo di progetto deve implementare: la quantità, nonché la tipologia di dati da memorizzare. Per il progetto in questione bisogna supportare il salvataggio di un mediocre ammontare di dati, garantire un efficiente accesso ai dati, controllare l'accesso di più utenti simultaneamente, storicizzare i dati acquisiti per avere la possibilità di stilare un sistema di feedback e di analisi degli sviluppi del progetto per permettere la valutazione di un fenomeno in espansione.

Dall'analisi dei dati in possesso, per garantire condivisione, persistenza e affidabilità del sistema e della base di dati, si è deciso di avvalersi di una base di dati gestita da un DBMS.

3.4.1. Analisi dei dati

Dall'analisi dei requisiti che il sistema dovrà implementare, è possibile derivare i concetti principali che modellano la realtà considerata. Sono state individuate 5 classi: gli utenti, gli annunci da loro inseriti, gli appuntamenti relativi agli annunci, i messaggi e lo storico delle collaborazioni, cioè i feedback di ogni scambio eseguito; di seguito, per ogni astrazione identificata, l'elenco delle principali caratteristiche che la descrivono, accompagnate da una breve descrizione, ove il significato non fosse chiaro a priori.

utente

Rappresenta la persona che, essendo socio attivo del progetto, avrà la possibilità di comunicare con gli altri utenti e collaborare con essi. Le caratteristiche specifiche da considerare sono:

Identificativo	Permette di identificare univocamente un utente nell'insieme degli utenti.
Dati anagrafici	Nome, cognome, indirizzo email, data di nascita, sesso.
Interessi	Rappresentano i dati che consentono di stilare un profilo dell'utente registrato.
Descrizione	Rappresenta il modo di avere un'istantanea dell'utente registrato.
Posizione	Le coordinate, latitudine e longitudine, che permettono di individuare la posizione geografica dell'utente
Crediti	Le ore che l'utente ha maturato effettuando gli scambi.

3. Analisi e progettazione

annuncio

Rappresenta l' oggetto che conterrà, suddivisi per richieste ed offerte, le inserzioni pubblicate dagli utenti che verranno lette e serviranno da motore per gli scambi all' interno della BdT.

Identificativo	Permette di identificare univocamente un inserzione da un' altra.
Identificativo Utente	Permette di identificare il creatore dell' inserzione.
Data creazione	Permette di memorizzare la data di creazione dell' annuncio.
Categoria	Utile a definire la tipologia dell' annuncio inserito, utile per essere utilizzato da filtro di ricerca.
Titolo	Per avere un' idea del contenuto dell' annuncio, senza la necessità di leggerlo per intero.
Descrizione	Il contenuto dell' annuncio.
Tipologia	Se l' annuncio è una richiesta o un' offerta.

messaggio

Rappresenta l' entità che consente agli utenti di comunicare tra di loro.

Mittente	Chi invia il messaggio.
Destinatario	Colui che leggerà il messaggio.
Data invio	La data di invio del messaggio in questione.
Oggetto	Il titolo visualizzato come anteprima del messaggio.
Corpo	Il testo del messaggio.
Cancellato	Campo che permette la gestione dell' eliminazione del messaggio.

evento

Rappresenta l' oggetto che permette all' utente di iscriversi ad un annuncio letto, attraverso questa entità un utente ha la possibilità di notificare, all' autore dell' annuncio la sua intenzione a collaborare con quest' ultimo in merito all' annuncio pubblicato.

3. *Analisi e progettazione*

Nome utente	Permette di definire chi si registra per lo scambio.
Nome destinatario	Colui che ha inserito l'annuncio, e deve essere notificato del fatto che un utente è interessato alla collaborazione.
Identificativo annuncio	Mette in relazione questa entità con l'entità annuncio per identificare univocamente l'evento all'annuncio corrispondente.
Data scambio	Permette agli utenti di accordarsi per effettuare la collaborazione.
Stato	Permette di indicare lo stato dell'evento, se è stato appena inserito o se è già stato letto e confermato dall'inserzionista.

feedback

Permette di tenere traccia delle collaborazioni effettuate e valorizzare una collaborazione mediante un voto ed una breve nota. Composto da:

Nome utente	Definire chi lascia il commento.
Nome destinatario	Definisce colui il quale riceve la valutazione in merito al lavoro eseguito.
Identificativo evento	Permette di identificare univocamente l'evento per il quale si sta lasciando il commento.
Voto	Un numero intero indice di gradimento del lavoro eseguito.
Nota	Un breve commento che descriva e giustifichi l'indice di gradimento.

3. Analisi e progettazione

3.4.2. Schema relazionale

Lo schema che segue modella la realtà appena descritta.

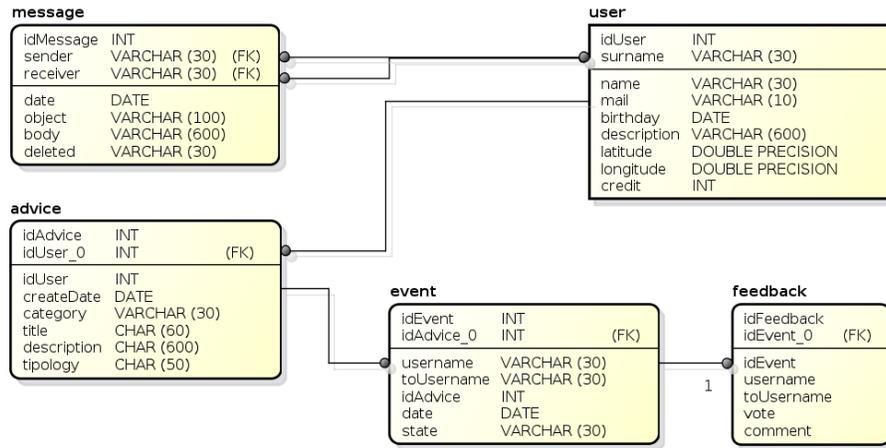


Figura 3.4.1.: Schema relazionale base di dati

In seguito ad una più accurata analisi si è deciso di eliminare la ridondanza che la tabella 'feedback' avrebbe creato, difatti un feedback deve essere memorizzato a seguito di un evento, che è stato precedentemente registrato ed eseguito. Si è dunque deciso di aggiungere gli attributi 'voto' e 'nota' alla tabella 'event' eliminando così la tabella 'feedback'.

Per quanto riguarda la tabella 'advice' si è deciso di aggiungere l'attributo 'username', che può comunque essere recuperato dalla relazione con la tabella 'user', ma in tale modo viene facilitato il recupero degli annunci quando si conosce soltanto lo 'username' di un utente, evitando così di effettuare una richiesta alla base di dati.

Di seguito lo schema relazionale rivisitato.

3. Analisi e progettazione

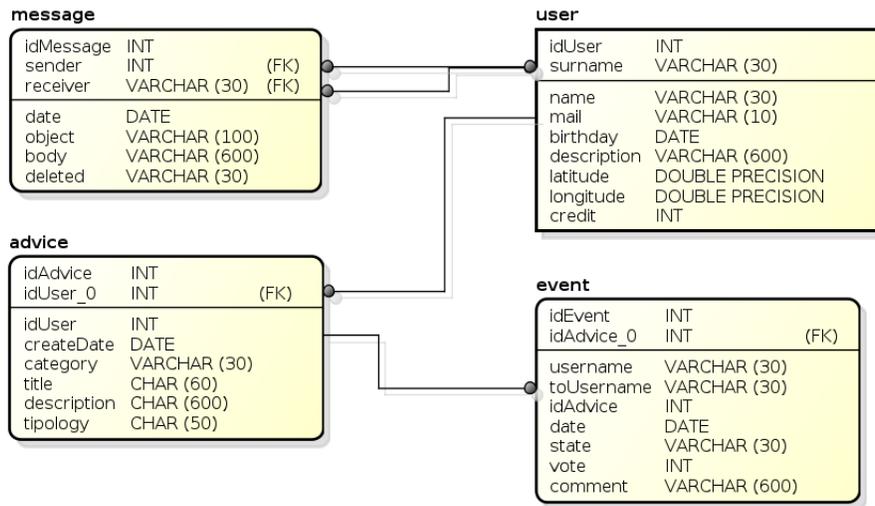


Figura 3.4.2.: Schema relazionale base di dati rivisitato

3.5. **Requisiti funzionali**

Di seguito vengono descritte le interazioni tra gli utenti ed il sistema attraverso i possibili casi d'uso, servendoci delle entità prima definite. Per ciascuna entità analizzata si ricorre ad un Sequence diagram che descrive l'interazione tra gli oggetti del sistema, rappresentando le comunicazioni tra di essi, secondo una linea temporale. Verranno prima delineate le operazioni principali eseguite da ogni entità, servendoci di un diagramma dei casi d'uso; poi verranno dettagliatamente analizzate le operazioni da implementare, rappresentando con un diagramma di sequenza, per ogni entità considerata, le operazioni che gli attori intraprendono, per soddisfare una determinata funzionalità.

Il seguente diagramma dei casi d'uso mostra le operazioni che un utente, loggato, può effettuare all'interno del sistema.

3. Analisi e progettazione

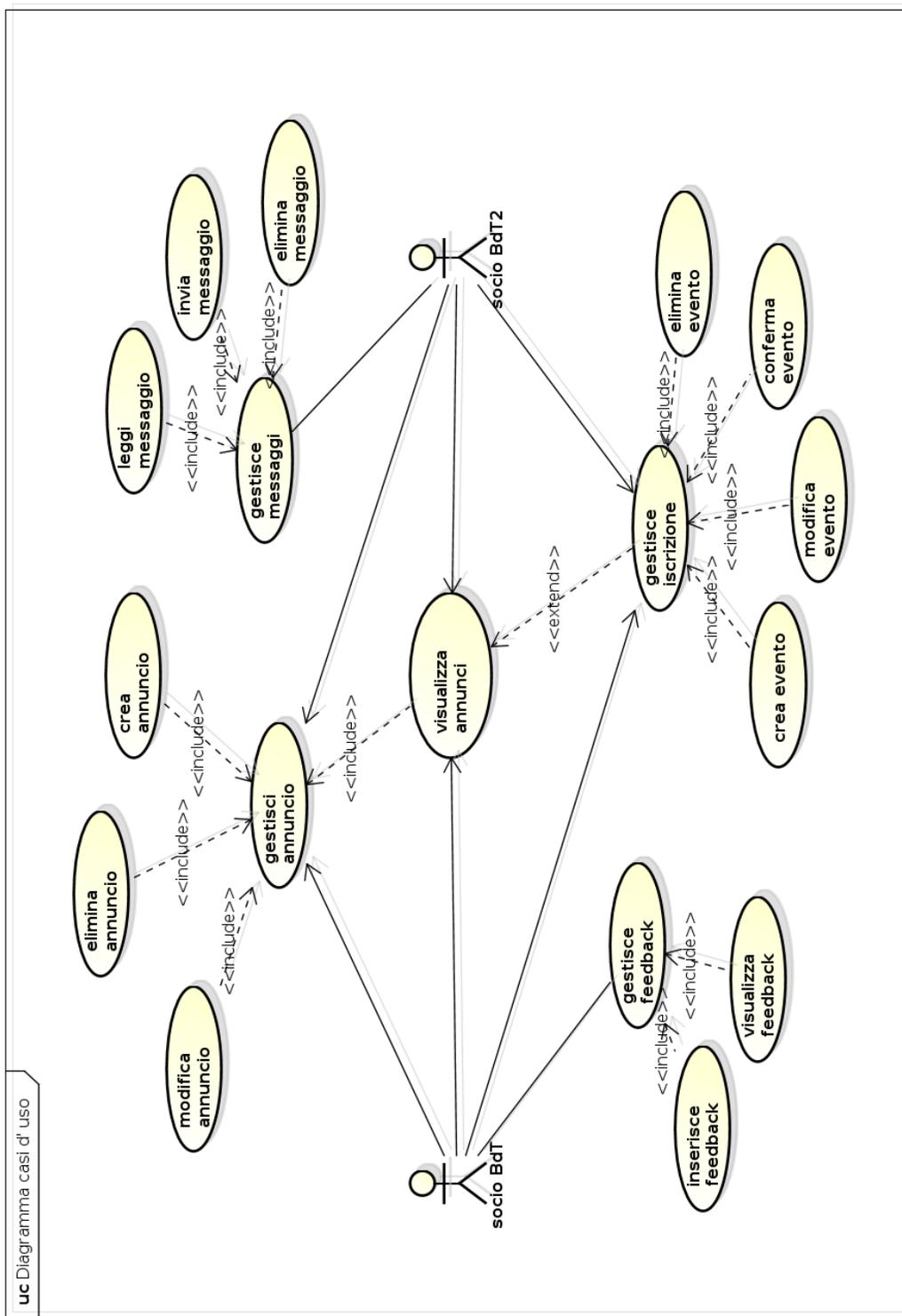


Figura 3.5.1.: Diagramma dei casi d' uso

3. Analisi e progettazione

Non sono state specificate le operazioni di registrazione nuovo account e login in quanto è il portale a gestirle.

3.5.1. Analisi delle funzionalità

Per ogni entità prima descritta vengono menzionate le operazioni disponibili e un diagramma di sequenza che mostra l'implementazione delle varie funzionalità:

utente

Le funzionalità che riguardano chi utilizzerà l'applicativo riguardano:

- Creazione nuovo account.
- Login all'applicativo.
- Modifica profilo personali (informazioni personali, posizione geografica)
- Visualizzazione profili utenti della comunità.
- Visualizzazione crediti e debiti maturati.
- Visualizzazione utenti nella mappa.

3. Analisi e progettazione

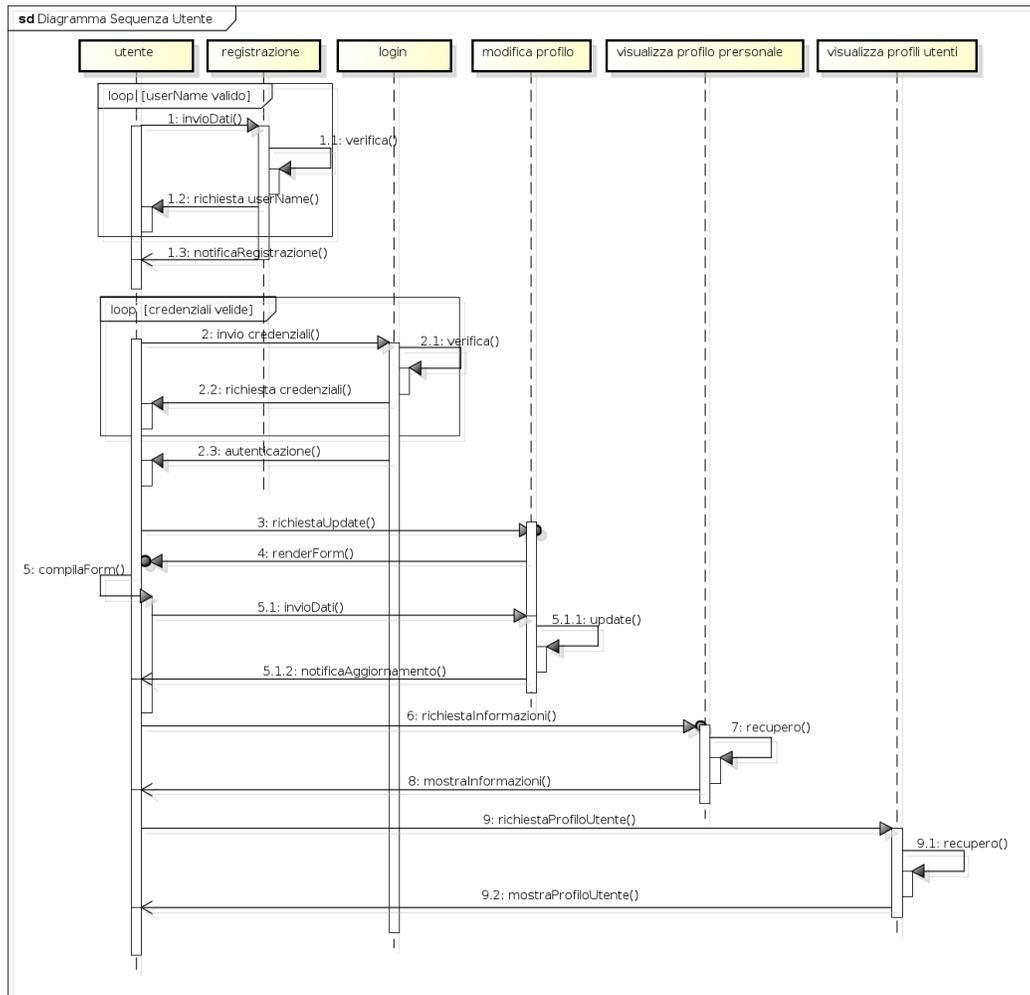


Figura 3.5.2.: Diagramma di sequenza, entità Utente

3. Analisi e progettazione

A seguire, tutte le operazioni descritte, hanno come prerequisito il fatto che l'utente sia registrato alla BdT, nei diagrammi di sequenza seguenti l'entità 'utente' è denominata 'socio'.

annuncio

Ogni utente dovrà avere la possibilità di inserire delle inserzioni che descrivano la tipologia di collaborazione che offrono o richiedono ai soci della BdT, le funzionalità riguardanti gli annunci riguardano dunque:

- Creazione nuovo annuncio.
- Modifica ed eliminazione annuncio precedentemente inserito.
- Visualizzazione annunci in base a filtri quali categoria oppure tipologia, quindi se l'annuncio è una richiesta o un'offerta.

3. Analisi e progettazione

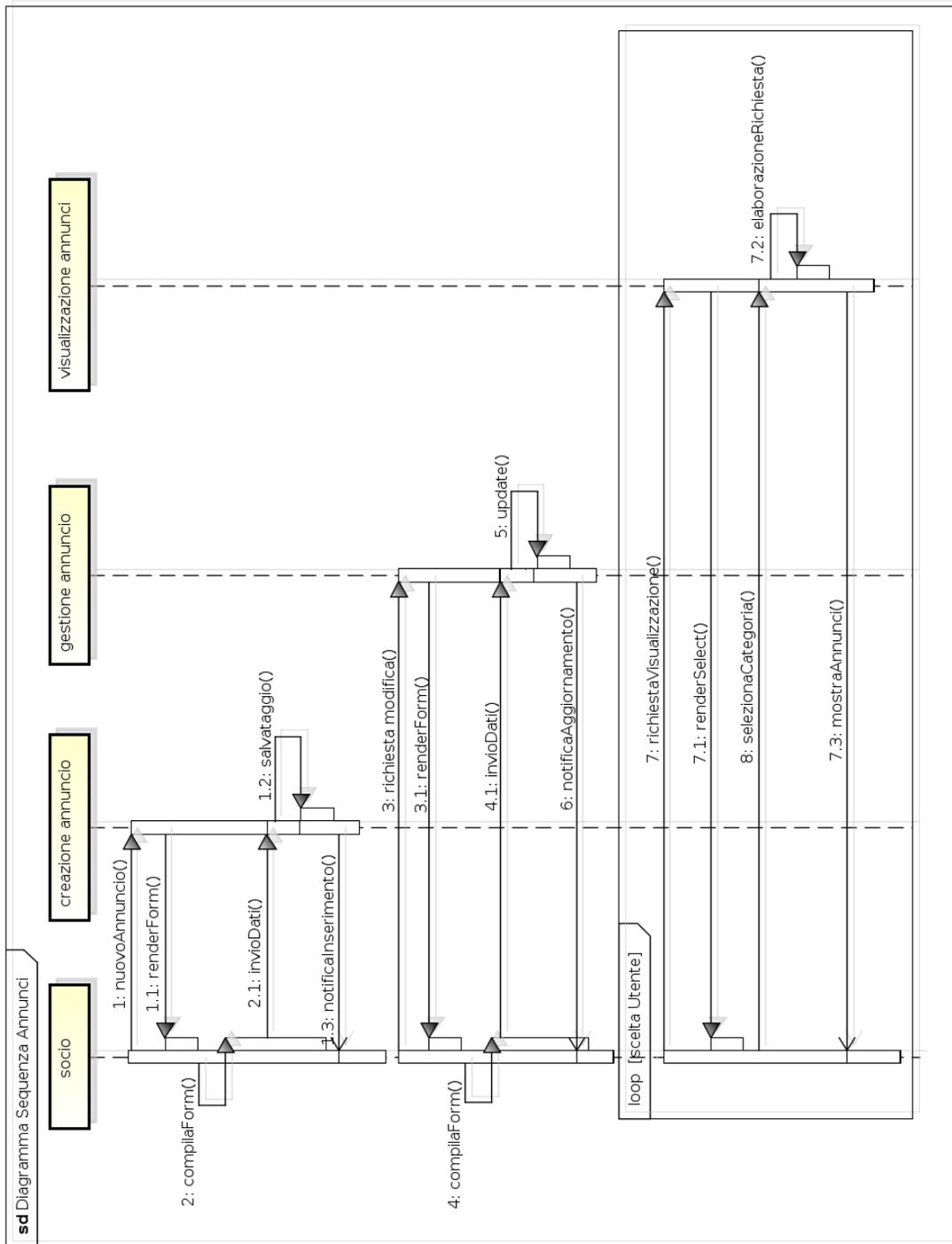


Figura 3.5.3.: Diagramma di sequenza, entità Annuncio

3. Analisi e progettazione

messaggio

L' applicativo deve consentire ai soci della BdT di comunicare, per far questo sarà necessario implementare un' entità 'messaggio' che dovrà consentire all' utente di effettuare le seguenti operazioni:

- Lettura messaggi ricevuti ed inviati.
- Invio messaggio ad un utente.
- Eliminazione messaggio.

3. Analisi e progettazione

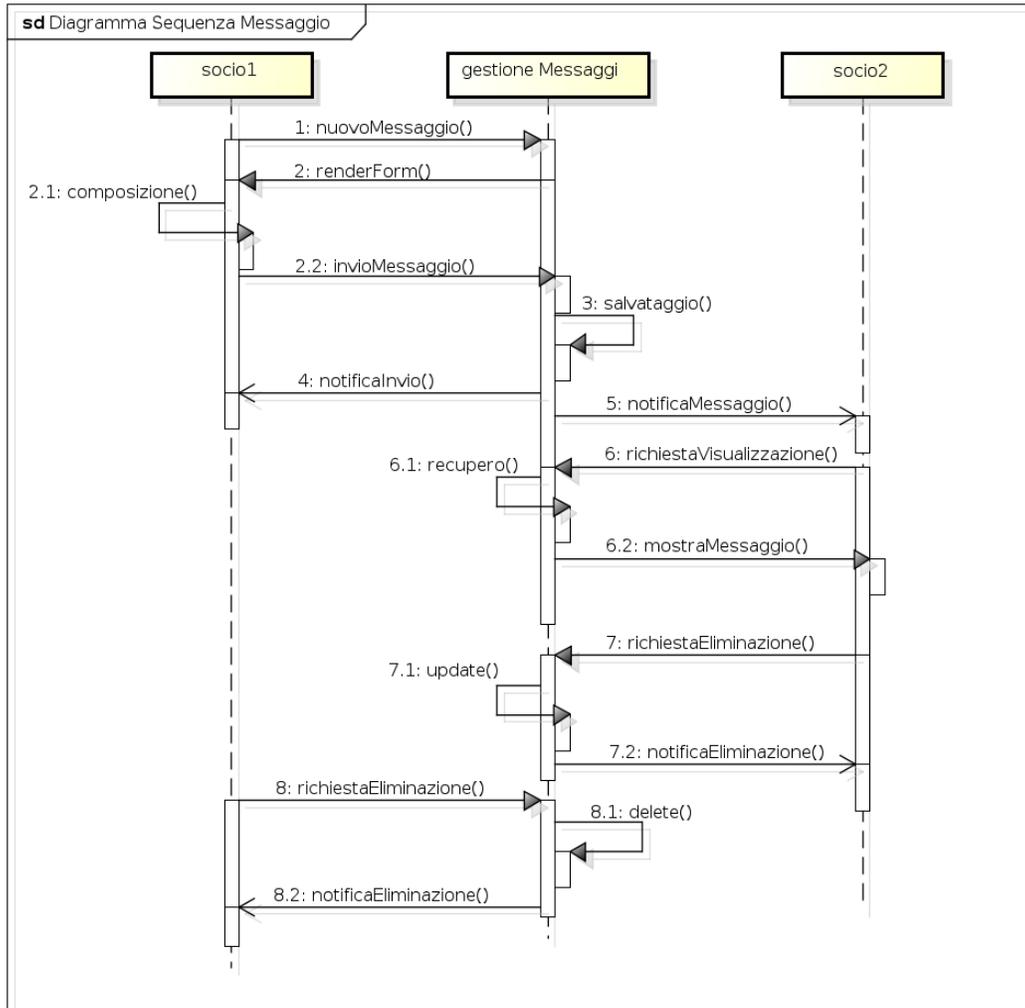


Figura 3.5.4.: Diagramma di sequenza, entità Messaggio

3. *Analisi e progettazione*

evento

Quando un utente è interessato ad un annuncio, dopo averlo letto tra la lista disponibile, nell' apposita sezione contenente la lista di tutti gli annunci dell' applicativo, o nelle pagine pubbliche di uno specifico socio, può notificare al creatore dello stesso il suo interesse al suddetto annuncio. La registrazione prevederà, quindi, l' immissione di una data in cui l' utente è disponibile ad effettuare la transazione, sia che si tratti di una richiesta, che di un' offerta, di modo che l' inserzionista, visualizzando questo interesse, abbia la possibilità di confermare o meno, in base alla sua disponibilità.

Dopo che un utente inserisce quindi un evento che riguarda uno specifico annuncio, le operazioni possibili sono:

- Modifica della data dell' evento, azione permessa sia al creatore dell' annuncio, che dopo aver ricevuto la notifica delle registrazione, abbia la necessità di modificare la suddetta data, sia a chi ha inserito l' evento, che vuole, successivamente all' inserimento, modificare la data precedentemente salvata.
- Conferma dell' evento inserito.
- Eliminazione dell' evento inserito.

3. Analisi e progettazione

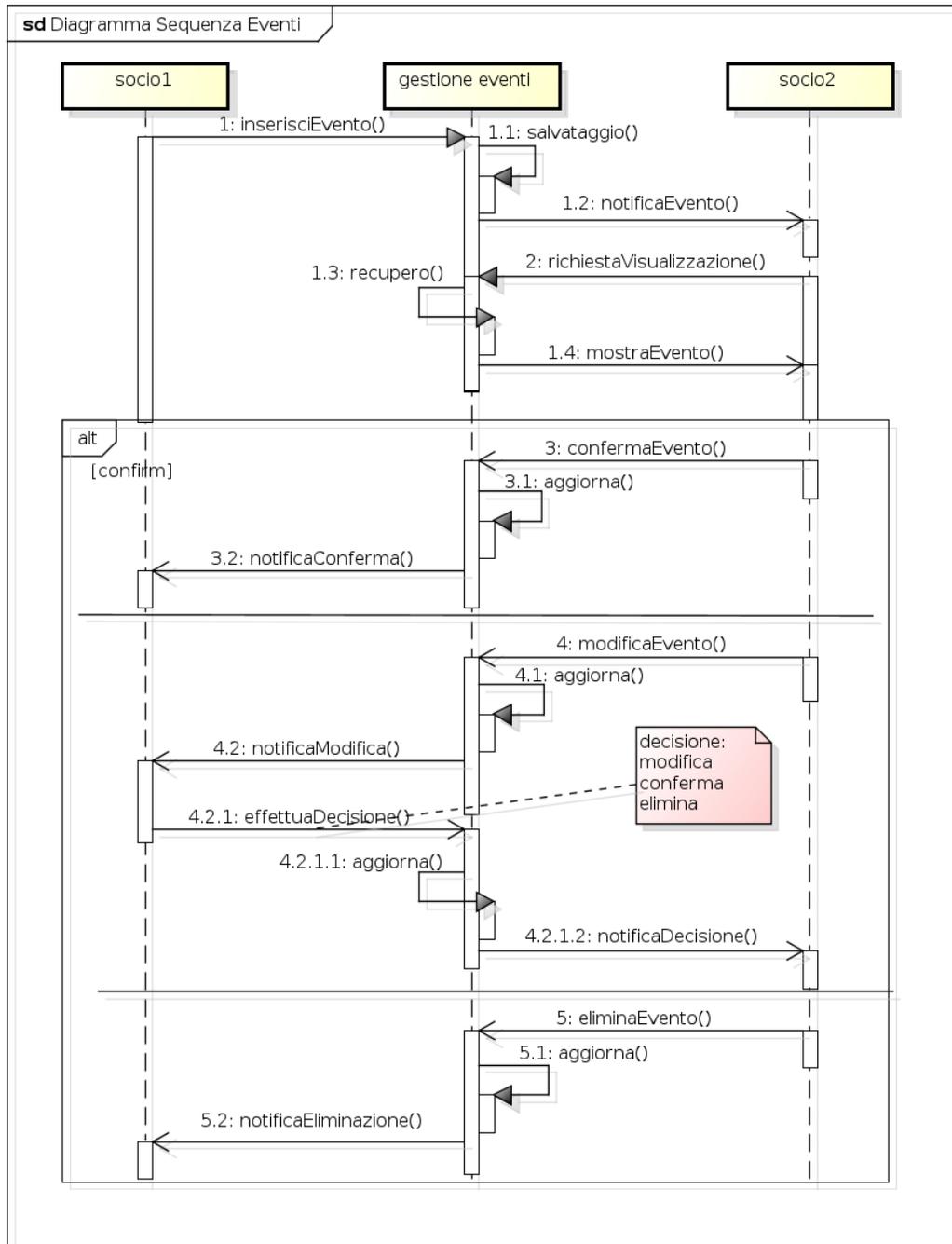


Figura 3.5.5.: Diagramma di sequenza, entità Evento

3. Analisi e progettazione

feedback

Dopo che due utenti hanno effettuato uno scambio, il socio che ha messo a disposizione il proprio tempo e le proprie abilità per aiutare l'altro socio, viene ricompensato con un credito pari al numero di ore effettivamente impiegate e con un commento, comprensivo di un voto che attesti la sua abilità, riguardo uno specifico campo di azione. Riguardo l'entità 'feedback' dunque, l'applicativo dovrà implementare le operazioni utili a:

- Lasciare voto e commento riguardo la transazione eseguita.
- Visualizzare lo storico di tutti i commenti riguardo un utente, quindi la lista dei feedback ricevuti dagli altri utenti, e la lista dei feedback lasciati agli utenti della BdT.

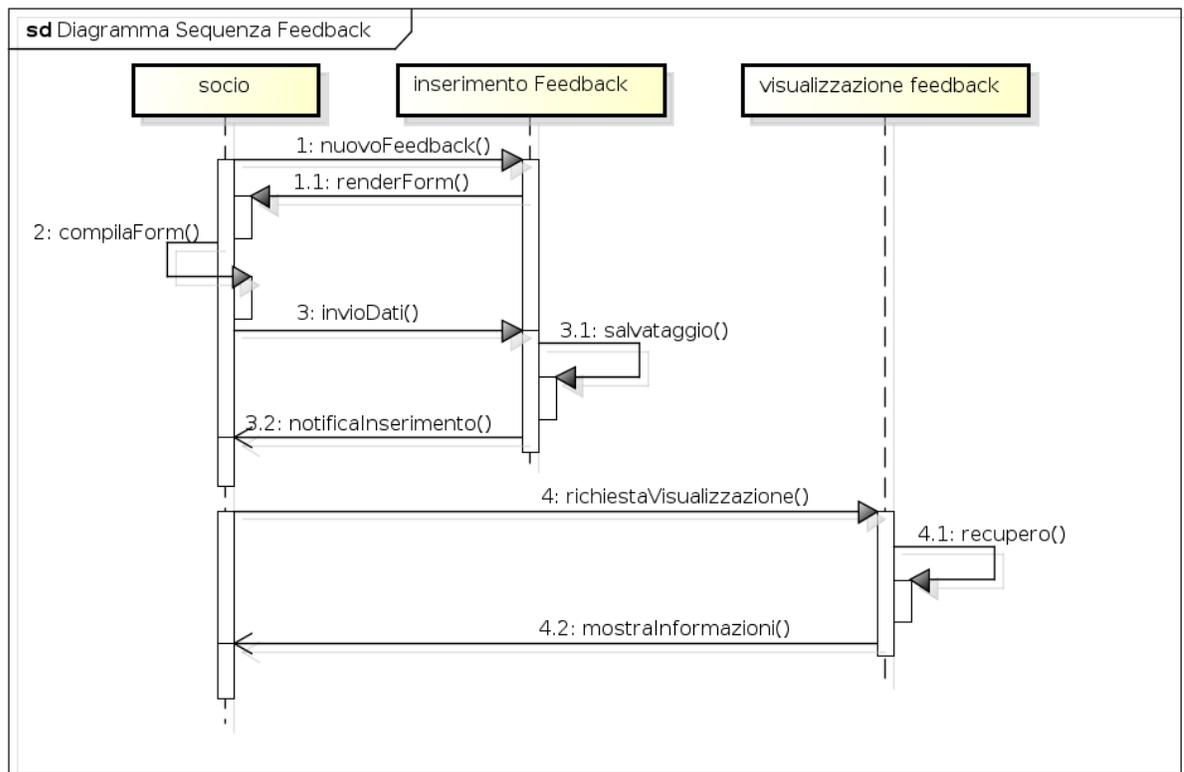


Figura 3.5.6.: Diagramma di sequenza, entità Feedback

4. Ambiente e strumenti di lavoro

Per lo sviluppo del progetto ho ricercato strumenti stabili, di facile utilizzo e gratuiti, che venissero incontro all' esigenza della gestione di una rete sociale basata sul concetto di community. Essendo un progetto che prevede sviluppi futuri l' idea di fondo era quella di avere un Content Management System, che potesse accogliere funzionalità in continuo sviluppo, magari create da diversi sviluppatori. Da qui la scelta del portale open source Liferay, essendo scritto principalmente in linguaggio JAVA (back-end) e JSP (front-end), e basato sulla filosofia open-source. Come DBMS, ho scelto MySQL, mentre per lo sviluppo delle portlet ,che il portale conterrà, mi sono orientato verso uno strumento innovativo, che si è dimostrato valido ed ha soddisfatto le mie aspettative, il framework VAADIN, che consente di creare, con un minimo know-how, salvo la conoscenza di java, rich text application complete e funzionali.

Nel corso di questo capitolo verranno presentati questi strumenti di sviluppo, prestando attenzione alle caratteristiche di tali strumenti inerenti le finalità progetto.

4.1. Liferay

Liferay portal è disponibile in due versioni, Liferay Portal Community Edition, versione gratuita, e Liferay Portal Enterprise Edition, versione commerciale che comprende supporto tecnico e che risulta più stabile rispetto la versione gratuita dovendo sottostare e superare ripetuti cicli di controllo della qualità. Il portale è distribuito sotto la licenza GNU ¹ e rispetta lo standard portlet JSR-286², essendo uno strumento open-source, presenta, a

¹La GNU Lesser General Public Licence è una licenza di software libero creata dalla Free Software Foundation e studiata quale compromesso tra la GNU General Public License e altre licenze non-copyleft. Fu scritta nel 1991 (aggiornata nel 1999 e nel 2007) da Richard Stallman, con l'ausilio legale di Eben Moglen.

²JSR 286 è la Java Portlet Specication v2.0 sviluppata sotto licenza JCP e creata in allineamento con la versione 2.0 di WSRP. E' stata sviluppata per migliorare la versione 1.0 (JSR-168). Tra le maggiori feature incluse vi sono: consentire intra portlet comuni- cation attraverso eventi e parametri pubblici di rendering; servire in modo dinamico le risorse generate direttamente attraverso le portlet; servire data AJAX o JSON attraverso le portlet; introdurre listeners e filters per le portlet.

4. Ambiente e strumenti di lavoro

differenza dei prodotti proprietari, alcune limitazioni e difficoltà legate principalmente alla quasi totale assenza di documentazione allineata alle versioni rilasciate. Questo limite, per quanto grande, può essere superato grazie alla disponibilità dei sorgenti del prodotto e di un valido forum on-line, consentendo, con un costo legato al tempo di acquisizione del know-how personale, di superare anche le situazioni più ostiche. Il prodotto difatti include una vasta scelta di “portlet” già sviluppate, sia dal team di sviluppo che dalla community, che consentono di ridurre notevolmente il tempo di “messa in opera” di un portale.

tra queste elenco troviamo:

- blog
- wiki
- chat
- forum
- sondaggi
- calendario eventi
- galleria immagini
- contenuti web
- gestione documenti
- gestione meetups

Incluse nell’ elenco soltanto le portlet di tipologia ‘social’ utili all’ implementazione del progetto, sono parecchie difatti le portlet che il portale mette a disposizione dell’ utente finale.

Caratteristica fondamentale di Liferay, il fatto che rende pratica la costruzione di siti web che presentano contenuti diversi a seconda che l’utente sia o no loggato, permettendo una diversa visualizzazione a seconda del ruolo dell’utente, consentendo agli utenti finali di definire autonomamente le pagine nonché di aggiungere dei contenuti utilizzando portlet predefinite o gadget. Gli utenti possono anche definire chi può accedere alle proprie pagine e ai propri contenuti.

Motivazioni dietro la scelta di questo strumento:

4. Ambiente e strumenti di lavoro

- creazione di pagine pubbliche e private per ogni utente.
- supporto a più Servlet Container (Apache TomCat, JBoss, GlassFish).
- possibilità di visualizzazione e utilizzo dell' applicazione da qualsiasi tipo di device, anche mobile.
- gestione automatica degli accessi.
- possibilità di definire nuove comunità e nuovi amministratori in caso di espansione del progetto in nuove località.
- utilizzo delle portlet sociali preinstallate.
- possibilità di aggiunta di nuove funzionalità, sotto forma di portlet.
- Personalizzare temi e layout.

4.1.1. I ruoli e i permessi

In Liferay la gestione degli utenti e dei rispettivi ruoli è dettagliata in quanto indispensabile per il pieno utilizzo delle potenzialità del portale, difatti definendo gli utenti, e assegnando loro specifici ruoli, è possibile controllare nel dettaglio l' interazione di ogni utente con i contenuti del portale.

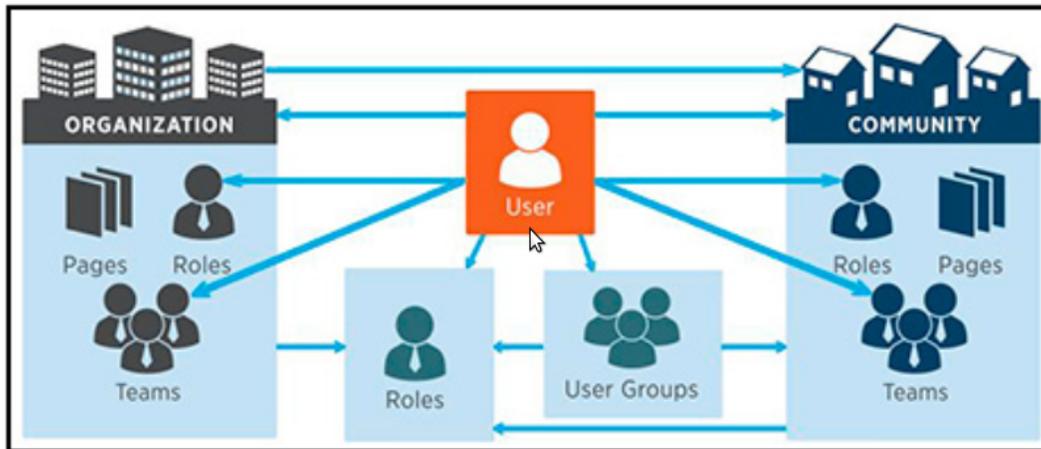


Figura 4.1.1.: Schema architettura ruoli in Liferay

4. Ambiente e strumenti di lavoro

L'immagine evidenzia il ruolo dell'utente che può, a differenza del visitatore comune, appartenere ad una comunità, ad una organizzazione, ad un gruppo di utenti, ad un ruolo o ad un team. Sia le 'community' che le 'organization' possono contenere pagine e gruppi di utenti, ma le organizzazioni, a differenza delle comunità, vengono utilizzate quando è necessario strutturare il portale secondo criteri gerarchici, utili nella creazione di intranet. Mentre le community che al momento della creazione possono essere definite come:

- Open, visibili ed accessibili per la registrazione da qualsiasi utente
- Restricted, nelle quali l'amministratore garantisce l'accesso
- Hidden, comunità Restricted particolari, nascoste e non vengono mostrate nell'elenco delle comunità del portale; sono particolari gruppi di utenti che condividono interessi comuni.

Ad una community corrispondono un insieme di pagine, a cui può essere associato un virtual host sia per la parte pubblica del portale che per quella privata. In pratica le comunità sono aggregazioni trasversali di utenti ai quali corrisponde un vero e proprio sito dedicato all'interno di un portale. In una singola istanza di Liferay possono coesistere quindi innumerevoli comunità ognuna con il suo virtual host, il che significa che ad una singola istanza di portale possono in realtà corrispondere innumerevoli siti distinti.

Un Gruppo di Utenti è un insieme di entità 'utente' che l'amministratore aggrega per scopi pratici di creazione e visualizzazione di contenuti. Sia utenti che gruppi di utenti possono far parte di una comunità o di un'organizzazione, così che quando una community possiede contenuti quali blog, wiki o forum, l'amministratore può facilmente assegnare ad un gruppo di utenti i permessi necessari alla gestione dei contenuti.

L'Utente è l'entità principale: dove per utente si intende un account di accesso al portale. L'utente può appartenere a una comunità, a una organizzazione, a un gruppo di utenti, a un ruolo o a un team. Prima di autenticarsi, ciascun utente è definito come guest ed ha un accesso limitato alle funzionalità implementate dal portale. Una volta autenticato, il client può avere determinati privilegi a seconda della comunità a cui appartiene, oppure in base all'organizzazione di cui fa parte.

I ruoli infine, individuano i permessi relativi ad ogni utente all'intero portale, nel prossimo capitolo verranno chiariti mediante un esempio pratico.

4.1.2. Creazione di una nuova community

Per la creazione di una nuova community Liferay mette a disposizione una portlet, 'comunità', che visualizza le comunità disponibili e dà la possibilità all'utente loggato di registrarsi ad

4. Ambiente e strumenti di lavoro

una comunità esistente, se essa è stata creata come comunità aperta, mentre rende possibile la creazione di una nuova comunità all'utente 'administratore'.

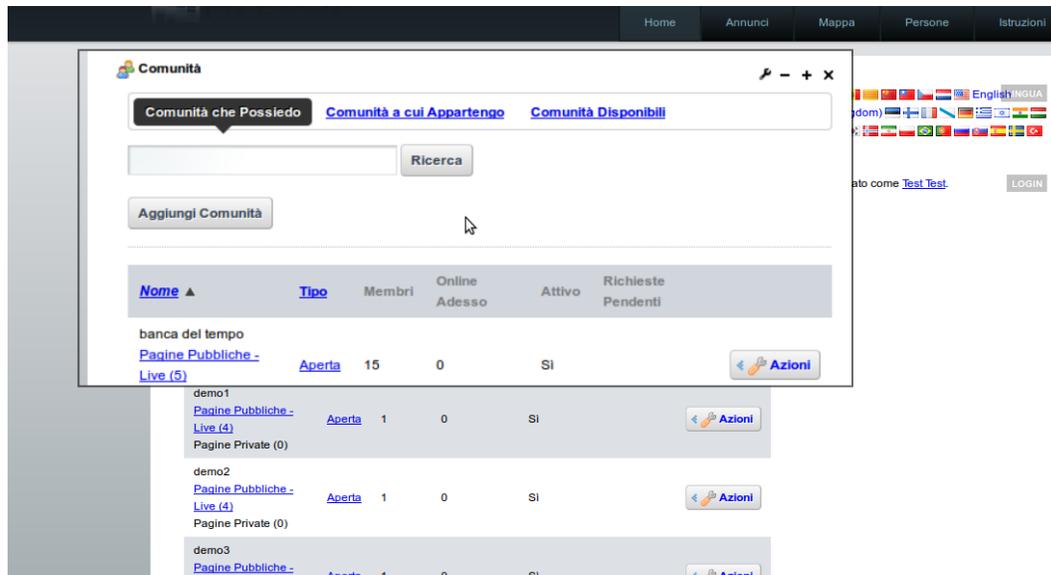


Figura 4.1.2.: Portlet , Particolare gestione community

La comunità 'banca del tempo' è stata creata; aggiungendo al file portal-ext.properties la riga:

```
admin.default.group.names = banca del tempo
```

Listing 4.1: Estratto file di configurazione di liferay

ogni nuovo utente sarà automaticamente aggiunto alla comunità "banca del tempo".

4.1.3. Pannello di controllo ed installazione plugin e nuovi temi

Dopo l'autenticazione come amministratore di sistema, con le credenziali username:"test" password:"test", è disponibile l'accesso al pannello di controllo. Molte delle proprietà configurabili con i file ".properties", con la versione Liferay 6.1 sono adesso disponibili nell'ambiente grafico del pannello di controllo. Sulla parte sinistra dello schermo è presente il menù suddiviso per categorie:

4. Ambiente e strumenti di lavoro

- Info utente: Gestione delle informazioni relative all' utente loggato.
- Comunità: Gestione della comunità; contenuti, aspetto grafico, e principali portlet inerenti alla comunità selezionata.
- Portale: Configurazione degli utenti, delle organizzazioni, delle comunità e ruoli ad essi annessi, nonché configurazioni generali inerenti al portale inclusi autenticazione, campi personalizzati e monitoraggio delle sessioni.
- Server: Gestione delle impostazioni del server su cui è installato Liferay, nonché gestione dei plugin e degli aggiornamenti.

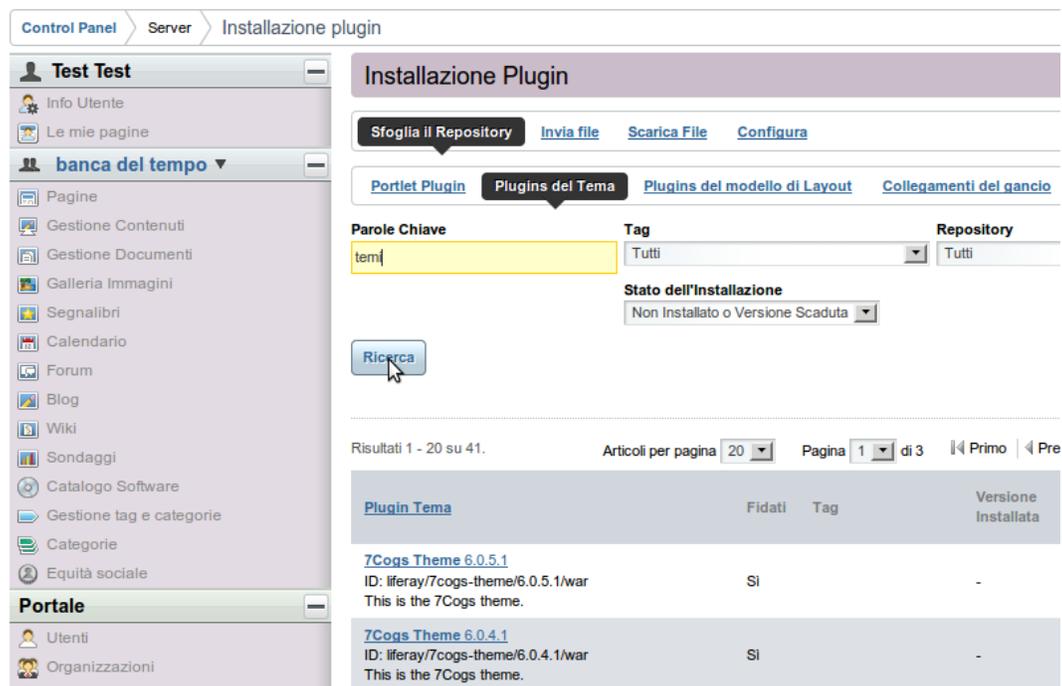


Figura 4.1.3.: Liferay, pannello di controllo

Può accedere al menù completo l' amministratore del sistema, gli utenti registrati hanno accesso soltanto alla sezione "info utente" salvo cambiamento dei relativi permessi attuabili nella sezione "Portale". La gestione dei temi delle pagine è accessibile nella sezione della relativa comunità; ad ogni comunità, come ad ogni pagina (pubblica o privata) può essere associato difatti un tema diverso.

L' installazione di nuovi temi e nuovi plugin, solitamente sotto forma di portlet, è accessibile nella sezione "Server", "installazione plugin". In questa sezione è possibile gestire i plugin

installati, effettuare aggiornamenti, ed installare nuovi plugin da file o dalla rete (presente un Form per la ricerca in base a parola chiave, tag o repository).

4.2. MySQL

MySQL è un sistema per la gestione di basi di dati relazionale (RDBMS - Relational database management system) che supporta gran parte della sintassi standard SQL e fruisce di varie interfacce per diversi linguaggi di programmazione attualmente esistenti. Questa comunicazione è resa possibile mediante l'uso dei driver proprietari: ciascuna applicazione web Java si interfaccia al database per mezzo del JDBC driver, chiamato anche MySQL Connector/J e usufruisce, nel caso specifico di liferay, della piattaforma Hibernate per l'amministrazione e la persistenza dei dati. L'insieme delle classi che offrono una connettività tra Java e un DBMS, implementando le interfacce dei package `java.sql` e `javax.sql`, prende il nome di driver JDBC; le principali interfacce sono: `Driver`, `Connection`, `Statement`, `PreparedStatement`, `CallableStatement`, `ResultSet`, `DatabaseMetaData`, `ResultSetMetaData`. Le classi implementative sono normalmente contenute in pacchetti ".jar" che lo sviluppatore dovrà caricare nella propria applicazione. Successivamente, per poter utilizzare i driver, si dovrà aprire una connessione con il database, creare uno `Statement` per interrogare la base dati e gestire i risultati ottenuti.

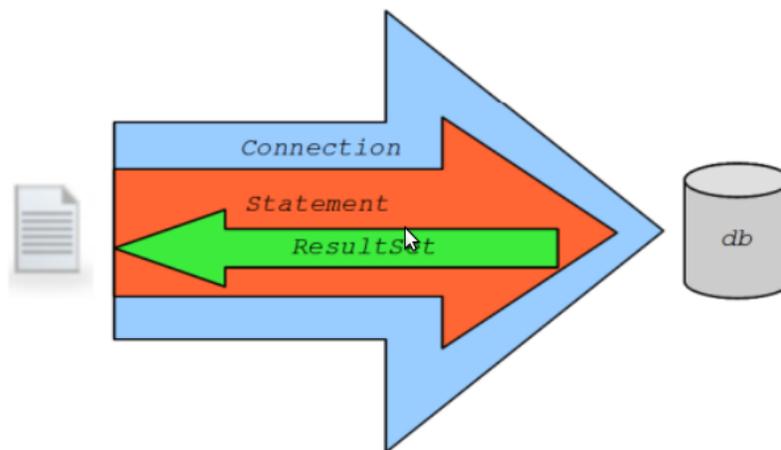


Figura 4.2.1.: Entità coinvolte nella connessione tra MySQL e una web Application

4.2.1. Integrazione con Liferay

Il portale liferay si appoggia, di default, al DBMS Hypersonic SQL Database; ma per un utilizzo più performante è meglio affidarsi a Database relazionali.

i passi da seguire per utilizzare il database con il portale liferay sono:

- aggiunta del MySQL Connector/J nel classpath del portale.
- dichiarazione parametri di connessione nel file di impostazione del portale: portal-ext.properties.
- definizione risorse nel file web.xml (cartella WEB-INF).
- definizione contesto nel file server.xml per ogni progetto in esecuzione.

4.2.2. Creazione tabelle

Al primo avvio del portale, viene eseguito un controllo sulla consistenza dei dati; se essi non vengono trovati il portale crea le tabelle nel database definito nel file “web.xml” con le righe:

```
<resource-ref>
<description>MySQL</description>
<res-ref-name>jdbc/portaldata</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```

Listing 4.2: Configurazione per la connessione al database

Dopo il primo avvio del portale, che impiegherà da 1 a 3 minuti, il database sarà pronto per l' utilizzo e conterrà alcuni dati utili al corretto funzionamento del portale.

Di default il portale conterrà alcuni contenuti di esempio, tra cui utenti e articoli blog e wiki, per un' installazione priva di dati superflui, prima di procedere con il primo avvio, rimuovere la cartella '7Cogs', situata dentro la cartella 'webapps' dal server.

4.2.3. Campi personalizzati

Le tabelle di cui necessita il portale sono create dal sistema al primo avvio, ma possiamo necessitare di ulteriori campi per una determinata entità; in questo caso, andando nel pannello di controllo come amministratori, nella sezione portale, possiamo aggiungere dei campi custom alle tabelle che ci interessano, tali campi saranno poi accessibili attraverso le classi e le interfacce create dal 'service builder' del portale; nel capitolo seguente verrà dettagliatamente descritto questo processo quando creeremo le tabelle per il sistema implementato.

4.2.4. Connessione al database

Il listato 4.3 mostra la porzione di codice utile alla connessione verso il database.

```
2 public Connection getConnection() throws NamingException, SQLException {
3     InitialContext cxt = new InitialContext();
4     DataSource ds = (DataSource)
5     cxt.lookup("java:/comp/env/jdbc/portaldata");
6     return ds.getConnection();
7 }
```

Listing 4.3: Codice per la connessione al database

Con metodo getConnection() instauriamo una nuova connessione con il database 'portaldata'. La figura 4.2.1 mostra le entità coinvolte.

Introduciamo qui anche il concetto JNDI; si tratta delle API che offrono servizi per identificare degli oggetti, o in generale, delle risorse con un nome, detto nome appunto JNDI. Le interfacce offrono nel loro insieme servizi di naming usando il paradigma Client/Server: il server offre i servizi JNDI e il client ne usufruisce per recuperare le risorse di cui ha bisogno. Il servizio è in esecuzione sull'Application Server JEE, questo permette all'Application Server di frapporsi tra l'applicazione e la connessione fisica al DBMS, aggiungendo nuovi servizi e funzionalità. Tra i più importanti vi è l'aggiunta di un pool connection che ottimizza e automatizza la connessione. Attraverso questo strumento, quando un'applicazione richiede una connessione a un database, l'Application Server verifica se vi sono connessioni aperte non utilizzate e se ciò si verifica, ritorna all'applicazione una di queste; quando l'applicazione non ha più bisogno di utilizzare la connessione, quest'ultima viene rilasciata con il metodo close() e resa disponibile ad altre applicazioni. L'invocazione di questo metodo da parte dell'applicazione non chiude fisicamente la connessione al DBMS, ma la rilascia all'Application Server, che potrà darla ad un altro Client.

4.3. Vaadin

Vaadin è una libreria per applicazioni web, Rich Internet Applications (RIA), dotata di una robusta architettura server-side che si basa su GWT per il rendering delle pagine. La maggior parte della logica di applicazione viene eseguita in modo sicuro sul server, qui la differenza sostanziale rispetto GWT. A questo si aggiunge la praticità nello sviluppo delle applicazioni; lo sviluppatore deve soltanto implementare la parte server delle applicazioni lasciando alla libreria il compito di creare le rispettive pagine http e jsp che verranno visualizzate dai browser, anche se rimane possibile modificare le pagine risultanti.

L'immagine mostra la struttura di questo strumento.

Nello specifico, il client-side-engine viene eseguito sul browser come un programma javascript, e attraverso il terminal adapter, comunica con il server-side-framework. La parte che risiede sul server (il server-side-framework) è composta da quelle entità, la logica applicativa e i componenti che costituiscono l'applicazione (che si occupano della logica di business e della logica dati), che gestiscono le interazioni tra l'utente e l'applicazione e offrono l'interfaccia applicativa all'utente. Ogni entità fornisce una determinata funzionalità ed è grazie a questa struttura che si può implementare un progetto, separando la gestione degli eventi e la creazione dell'interfaccia, dalla gestione dei temi e di altre risorse complementari (fogli di stile). Questa impostazione permette l'implementazione di componenti modulari, separabili e riutilizzabili in uno stesso progetto o in altri.

Nel prossimo capitolo verrà mostrata in dettaglio l'implementazione di un componente per la creazione di un Form. Adesso si chiariscono le modalità e i metodi di funzionamento della libreria all'interno del portale scelto.

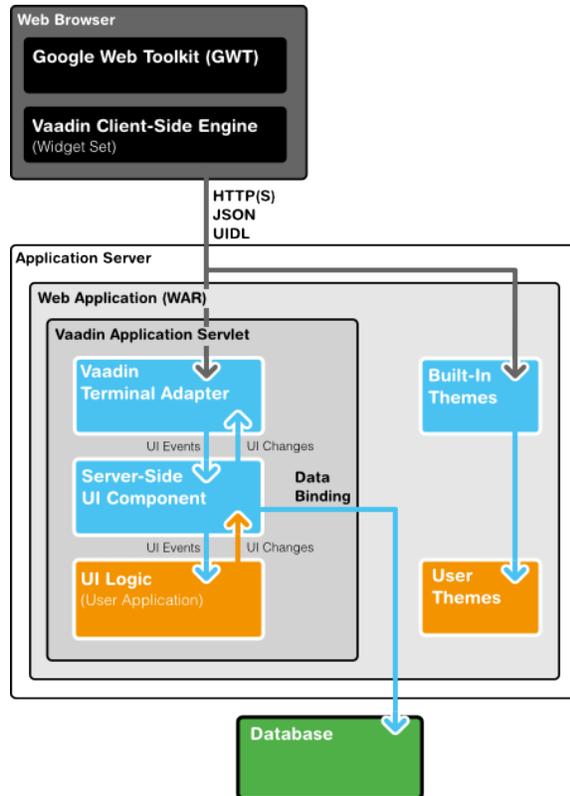


Figura 4.3.1.: Architettura vaadin

4.3.1. Integrazione con Liferay

Per utilizzare la libreria all' interno del portale Liferay è necessario:

- aggiungere la libreria 'vaadin.jar' al classpath di liferay.
- scompattare il contenuto della libreria nel context del portale.
- specificare nel file di configurazione(portal-ext.properties), il path in cui risiede la cartella appena scompattata.

```
vaadin.resources.path =/html  
vaadin.theme=reindeer  
vaadin.widgetset=com.vaadin.portal.gwt.PortalDefaultWidgetSet
```

Listing 4.4: File di configurazione per vaadin

nei parametri sopra indichiamo a liferay dove risiede la cartella di vaadin, il tema di default da utilizzare, e il widget set utilizzato dalle nostre portlet. La cartella estratta nel context di Liferay(/html), contiene:

- themes: contenente i temi di default ed eventuali temi personalizzati.
- widgetsets: contenente il widget set(DefaultWidgetSet) ed eventuali widget set ricompilati(PortalDefaultWidgetSet).

Il widget set scompattato, nel caso in cui aggiungessimo nuove funzionalità(add-ons di vaadin), deve essere ricompilato. E' consigliabile eseguire questo processo in locale; è utile a questo scopo un plugin, 'liferay control panel', scaricabile dall' area add-ons di vaadin; una volta installato nel portale, per mezzo del file ".war" contenuto in esso, è possibile aggiungere la portlet alla pagina, potremo cosi' gestire le librerie che utilizzeremo e ricompilare successivamente il widget set.

4. Ambiente e strumenti di lavoro

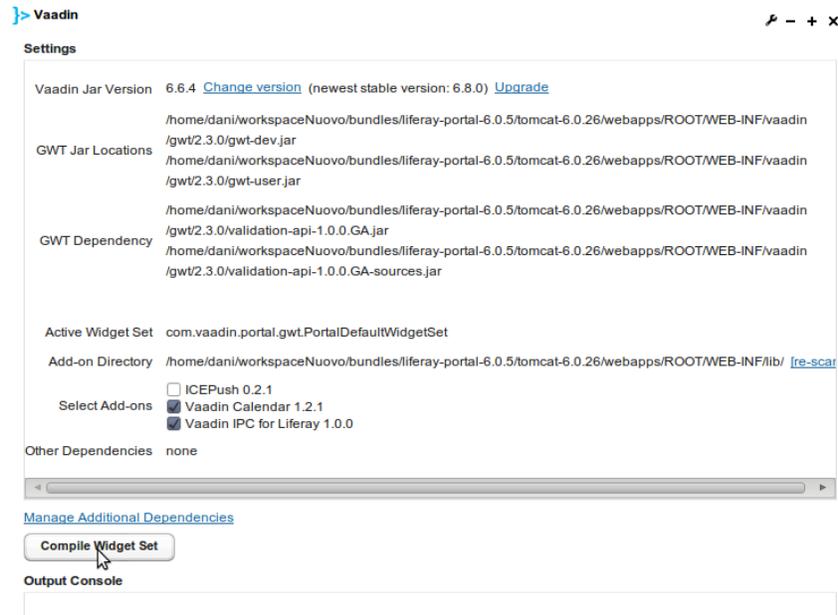


Figura 4.3.2.: Portlet , liferay control panel

Come mostra l'immagine attraverso questa portlet è possibile gestire le cartelle in cui risiede la libreria di vaadin e le risorse GWT da essa utilizzate. Ogni qualvolta dovessimo aggiungere un nuovo plugin basterà aggiungerlo a quelli presenti attraverso questo strumento e ricompilare il widget set, che verrà salvato nella cartella prima indicata.

4.3.2. Gestione dei temi

Vaadin separa l'aspetto dell'interfaccia utente dalla logica applicativa utilizzando i temi. I temi includono tutte quelle risorse tra cui fogli di stile CSS, HTML, o layout e immagini. La libreria mette a disposizione dello sviluppatore alcuni temi standard, la porzione di codice prima esposta indica a Liferay di utilizzare il tema "reindeer" di default, ma all'interno della stessa applicazione si possono utilizzare svariati temi.

Per creare un nuovo tema è conveniente seguire la gerarchia utilizzata nei temi standard di Vaadin. Una volta create le risorse necessarie, esse dovranno essere inserite in una cartella specifica, il nome della cartella sarà il nome del nuovo tema, la cartella può quindi essere collocata all'interno della cartella "themes".

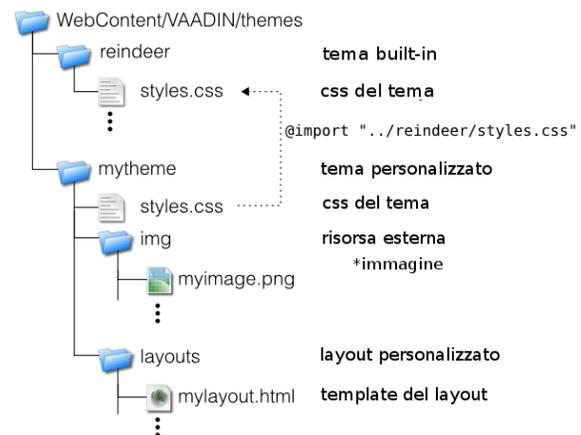


Figura 4.3.3.: Architettura cartella temi di Vaadin

Una volta creato il tema, che può comunque ereditare uno dei temi di default tramite la clausola di import, sarà possibile utilizzarlo all'interno del codice con il metodo `setTheme("temaCustom")`. Vaadin offre inoltre la possibilità di settare l'aspetto grafico di ogni singolo componente dell'interfaccia.

- `setTheme("nomeComponente")`: sovrascriverà le proprietà relative ad un componente.
- `addTheme("nomeComponente")`: aggiungerà delle proprietà ad un componente conservando le proprietà iniziali.

Con l'utilizzo di questi metodi è possibile assegnare a componenti inseriti nella stessa pagina, diversi aspetti grafici.

Un esempio di foglio di stile:

```
1 @import url(../reindeer/styles.css);  
  
3 /* Customizzazione Table */  
   .v-table-header-wrap { height: 20px;  
5 border: none;  
   border-bottom: 1px solid #555;  
7 background: transparent url(images/table-header-bg.png) repeat-x;  
   }  
9 .v-table-caption-container { font-size: 11px;  
   color: #000;  
11 font-weight: bold;  
   text-shadow: #fff 0 1px 0;  
13 padding-top: 1px; }  
   .v-table-body { border: none; }  
15 .v-table-row-odd { background: #f1f5fa; }  
   .v-table-row:hover { background: #fff; }  
17 .v-table-row-odd:hover { background: #f1f5fa; }  
   .v-table .v-selected { background: #3d80df; }  
19 .v-table-cell-content { padding: 2px 6px 2px 3px; line-height: normal; font-size: 85%; }
```

Listing 4.5: Css tema di Vaadin

Questo foglio di stile sovrascrive o arricchisce il componente 'table' del tema "reindeer", che è stato importato ad inizio pagina. Utilizzando il metodo `addTheme("table")` è possibile infatti conservare le impostazioni di stile delle tabella secondo il tema "reindeer", aggiungendo le proprietà espresse.

4.3.3. Add-ons

Gli add-ons sono delle librerie aggiuntive che la community di vaadin mette a disposizione degli utenti iscritti. Per vedere pubblicata una propria libreria è necessario, una volta membro della community, definire il tipo di libreria sviluppata, se si tratta di un componente, un tema, o uno tool di sviluppo, e a seguito di un testing sul prodotto e la composizione di una quanto più accurata documentazione, è possibile distribuirla in formato ".jar" o ".zip". Per l' utilizzo di questi plugin è necessario aggiungere l' archivio ottenuto al classpath del Progetto e in seguito alla ricompilazione del widget set, come mostrato prima, è possibile usare la libreria.

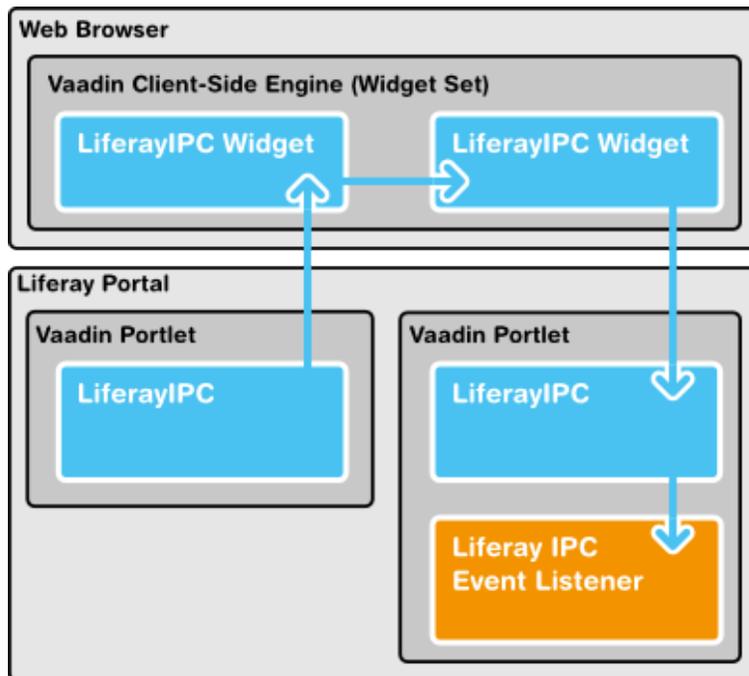


Figura 4.3.4.: Architettura add-ons liferay-ipc

4.3.4. Gestione degli eventi con la libreria liferay-ipc

Nel progetto si è spesso presentata la necessità di dover aggiornare il contenuto di un componente all'interazione di un utente, come ad esempio l'inserimento di un nuovo dato, con un nuovo contenuto. Per evitare il refresh dell'intera pagina, perdendo così vantaggi derivanti dall'uso della tecnologia ajax, la community di vaadin mette a disposizione dei suoi utenti un add-on che permette di effettuare questa operazione in assoluta facilità. La soluzione messa a disposizione da vaadin, basata sul pattern Observer, permette di comunicare tra le portlet sul lato server utilizzando un meccanismo di server-push per aggiornare il client-side.

Per implementare questo meccanismo, dopo aver aggiunto la libreria 'vaadin-ipc-for-liferay' al classpath del progetto, è necessario inviare l'evento(il messaggio) desiderato mediante il metodo `sendEvent()`. Le portlet che ricevono l'evento(il messaggio) dovranno registrare un listener con il metodo `addListener()`; quindi ogni qualvolta venga inviato un evento tutte le portlet in ascolto potranno, indipendentemente l'una dalle altre, eseguire il codice interno al proprio listener.

4.4. GoogleMap

Google Maps è l'insieme dei servizi offerti da google che, in seguito al rilascio della versione javascript API V3, ha arricchito e semplificato l'uso di tale tecnologie agli sviluppatori di pagine web. La georeferenziazione, le immagini dai satelliti, le operazioni di zoom e di calcolo delle distanze, la creazione automatica dei percorsi, l'inserimento di pop-up informativi, richiedono, con questa versione, un bassissimo know-how e un numero esiguo di righe di codice, rispetto la versione precedente, che permettono allo sviluppatore di creare sui siti web, così come sui dispositivi mobili, applicazioni robuste, pratiche e produttive.

```

1 <script type="text/javascript">
  <src="http://maps.google.com/maps/api/js?sensor=false">
3 </script>
4 <script type="text/javascript">
5 function initialize(){
6   var latLng = new google.maps.LatLng(44.4586384617695,11.3109901532338);
7   var myOptions = {
8     zoom : 8,
9     center : latLng,
10    mapTypeId : google.maps.MapTypeId.ROADMAP}
11  }
12  mymap = new google.maps.Map(document.getElementById("map"), myOptions); }
13 </script>
14 <html>
15 <body onload="initialize()">
16   <div id="map" style="width: 600px; height: 400px">
17 </div>
18 </body>
19 </html>

```

Listing 4.6: Codice per la visualizzazione di una mappa

Si è voluto inserire il codice necessario alla visualizzazione di una mappa, per mostrare la semplicità di utilizzo di questo servizio. Dopo il link al codice Google map è necessario implementare una funzione(javascript) che, definite le variabili utili alla visualizzazione della mappa, in questo caso il punto centrale e il tipo di mappa(stradale), le associ all'elemento mappa 'map'. A questo punto è possibile collocare la mappa sulla pagina con l'elemento prima definito ed eseguire la funzione al caricamento della pagina.

5. Implementazione e integrazione con il portale

Avendo definito il modello di dominio e l' ambiente su cui poggia il sistema, descriviamo adesso la struttura dell' applicativo, la base di dati, che andremo ad integrare con liferay, e le portlet che implementano le funzionalità offerte dal sistema.

5.1. Architettura progetto

Per l' implementazione del progetto si è adottata una strutturazione seguendo linee guida che potessero quanto più sia standardizzare il modello di sviluppo del progetto, sia favorire sviluppi futuri. L'impiego di un adeguato pattern porta numerosi vantaggi tra cui:

1. Incrementa il riutilizzo del codice.
2. Facilita il lavoro in team e la pianificazione del progetto, dividendo quest'ultimo in componenti indipendenti delegabili a gruppi di lavoro differenti.
3. Aiuta la manutenzione del codice, grazie anche all'abbassamento della curva di apprendimento (per i progetti più complessi).
4. Aumenta la flessibilità delle applicazioni e incrementa la scalabilità.

5.2. Struttura progetto

Di seguito il diagramma di comunicazione mostra le relazioni tra i moduli che compongono il progetto.

5. Implementazione e integrazione con il portale

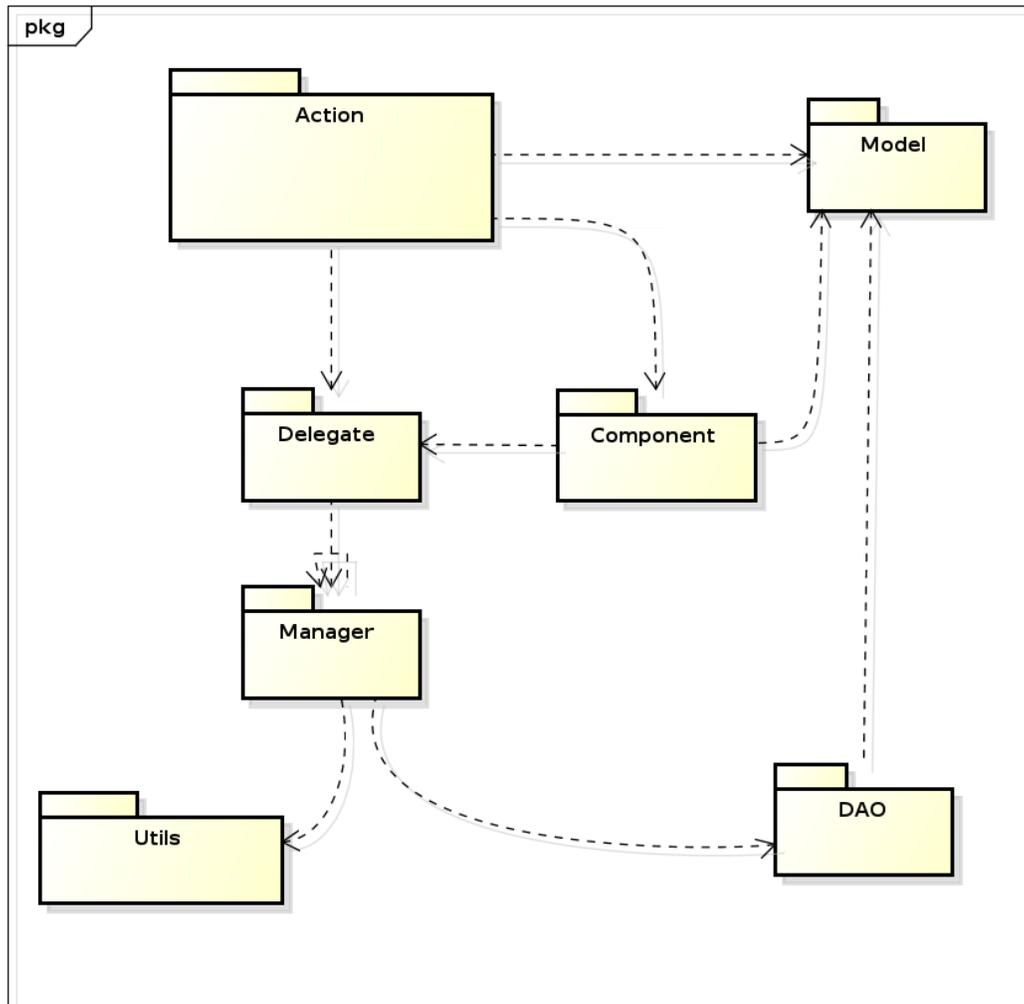


Figura 5.2.1.: Diagramma di comunicazione , struttura applicazione

Si è scelto di strutturare il progetto dividendolo in moduli facilmente scomponibili e modificabili. In breve l'analisi dei moduli.

'Action' è il package che contiene le classi java dedicate a visualizzare l'interfaccia per l'utente e gestire le interazioni tra l'utente e l'applicativo, in esso sono contenute tutte quelle classi che si servono:

dell'interfaccia 'businessDelegate', contenuta nel package 'delegate', per la comunicazione con la base di dati.

5. Implementazione e integrazione con il portale

delle classi contenute nel package 'Component', per la costituzione degli oggetti e delle entità che compongono l'interfaccia per l'utente.

Le classi contenute nel package 'Action' rimangono all'oscuro dei metodi necessari alla connessione con la base di dati e alla composizione dello specifico componente da aggiungere alla pagina, veicolando solamente questi dati e occupandosi dell'interazione tra l'utente e il sistema.

Il package 'Delegate' contiene l'interfaccia 'BusinessDelegate', nella quale sono centralizzati i dettagli dell'implementazione del livello business. Essa si serve delle classi contenute nel package 'Manager', per recuperare le liste di oggetti dalla base di dati.

Le classi contenute nel package 'Manager' comunicano con le classi facenti parte il package 'DAO', inviando la connessione fornita dalla classe 'DBConnector'(package 'utils') e i parametri per il recupero dei dati, queste classi si occupano di restituire al BusinessDelegate, e quindi alle classi 'Action' i dati recuperati dalla base di dati.

Le classi contenute nel package 'models' contengono gli oggetti che modellano la realtà considerata, le classi del package 'DAO' si occupano invece del recupero, inserimento e aggiornamento dei dati da e verso la base di dati.

Di seguito un'immagine utile a chiarire quanto appena detto.

5. Implementazione e integrazione con il portale

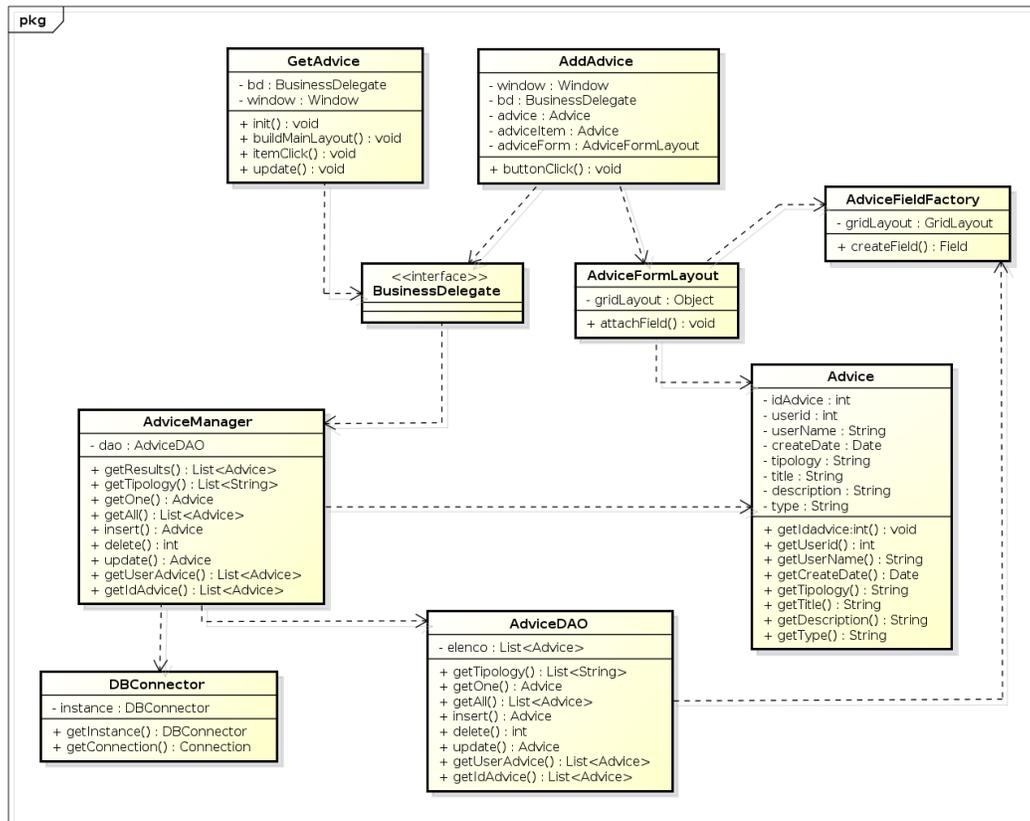


Figura 5.2.2.: Diagramma delle classi , struttura progetto

L'immagine mostra il diagramma delle classi "GetAdvice" e "AddAdvice", e delle classi ed interfacce con cui interagiscono per visualizzare gli annunci presenti nella base di dati, e inserire un nuovo annuncio.

5.3. La base di dati

Come discusso prima, al primo avvio del portale liferay vengono create le tabelle utili alla gestione e all' utilizzo del portale. Con le sue 168 tabelle liferay gestisce tutte le funzionalità, dal sistema di autenticazione degli utenti, all' interfaccia grafica, la gestione dei ruoli e il salvataggio dei contenuti che gli utenti possono creare e condividere all' interno del portale. Analizzare tutte le tabelle esula dalla finalità dell' elaborato, quindi verranno descritte le relazioni con le tabelle create per ' implementazione dell' applicativo, dopo aver creato tutte le tabelle utili allo stesso.

5.3.1. Integrazione con Liferay

Durante la fase di progettazione è stata definita, assieme le altre, l' entità utente, liferay contiene la tabella 'user_' che, opportunamente referenziata, sarà utilizzata dal nostro applicativo. Descriviamo prima il processo di creazione delle tabelle per dare in seguito la struttura risultante della base di dati.

5.3.2. Creazione tabelle mediante service builder

Il "Service Builder" è un generatore di codice che produce, organizzate in una precisa struttura in livelli, le classi e le interfacce utili a gestire i dati su tabelle custom. Mediante un' opportuna configurazione dei file da esso utilizzati, è possibile creare tabelle personalizzate e metodi utili a inserimento, lettura, cancellazione e aggiornamento dati. Di seguito vengono mostrati i passi necessari alla creazione della tabella 'message', le altre tabelle vengono create con lo stesso procedimento.

Dopo aver creato un progetto, è necessario creare il file service.xml all'interno della cartella "WEB-INF" del progetto in questione; esso contiene, espresse tramite la sintassi DTD, la dichiarazione delle tabelle e dei relativi attributi da creare. Di seguito il file service.xml per la creazione della tabella 'message'.

```
1 <!DOCTYPE service-builder PUBLIC "-//Liferay//DTD Service Builder 6.0.0
  //EN" "http://www.liferay.com/dtd/liferay-service-builder_6_0_0.dtd">
3 <service-builder package-path="it.social.timeBonk">
  <namespace>Timebonk</namespace>
5 <entity name="message" local-service="true" remote-service="true">
  <!-- PK fields -->
7 <column name="idmessage" type="int" primary="true" />
```

5. Implementazione e integrazione con il portale

```
<!-- Group instance -->
9 <column name="object" type="String" />
  <column name="send" type="String" />
11 <column name="receiver" type="String" />
  <column name="Data" type="Date" />
13 <column name="body" type="String" />
  <column name="sender" type="Collection" entity="User" mapping-key="screenName" />
15 </entity>
</service-builder>
```

Listing 5.1: File configurazione , service.xml per Service Builder

Descrivere il funzionamento di questo generatore esula dagli obbiettivi della tesi, a titolo esplicativo sono da notare i tag:

- local-service="true" remote-service="true": Permettono di creare, rispettivamente, interfacce locali e remote.
- entity name: Nome della tabella.
- coloumn name: Nome dell' attributo.
- column name="sender" type="Collection" entity="user_" mapping-key="screenName": Serve a specificare una relazione 1-n tra la tabella 'message' e 'user_'. In questo modo il Service Builder genera automaticamente i metodi per recuperare una lista di messaggi dato lo screenName di un user_, poiché è quella la colonna che usiamo per la relazione nell'attributo mapping-key. A titolo esplicativo, è possibile esprimere anche relazioni n-n, invece di mapping-key si usa l'attributo mapping-table.

Creato il file "service.xml" è possibile invocare il target build-service dell' ant, che si occuperà di creare, all' interno della cartella 'src' e della cartella 'lib' del progetto, le classi e le interfacce relative alla tabella appena creata; mentre al riavvio del portale verrà creata la tabella nel database del portale.

Liferay mette a disposizione dello sviluppatore una serie di interfacce utili alla gestione dei dati, senza la necessità di creare una nuova connessione al database, dato che il portale rimane sempre in comunicaione con esso per svolgere le operazioni di routine, più avanti, nel corso del capitolo, quando si introdurranno le 'expando table', verranno mostrati alcuni metodi utili al recupero dati.

5. Implementazione e integrazione con il portale

5.3.3. Campi personalizzati ed expando table

Liferay mette a disposizione dello sviluppatore un' importante feature, la possibilità di aggiungere alle tabelle presenti nel sistema campi personalizzabili, e offre allo stesso tempo metodi utili al recupero di tali dati nonchè al salvataggio di essi.

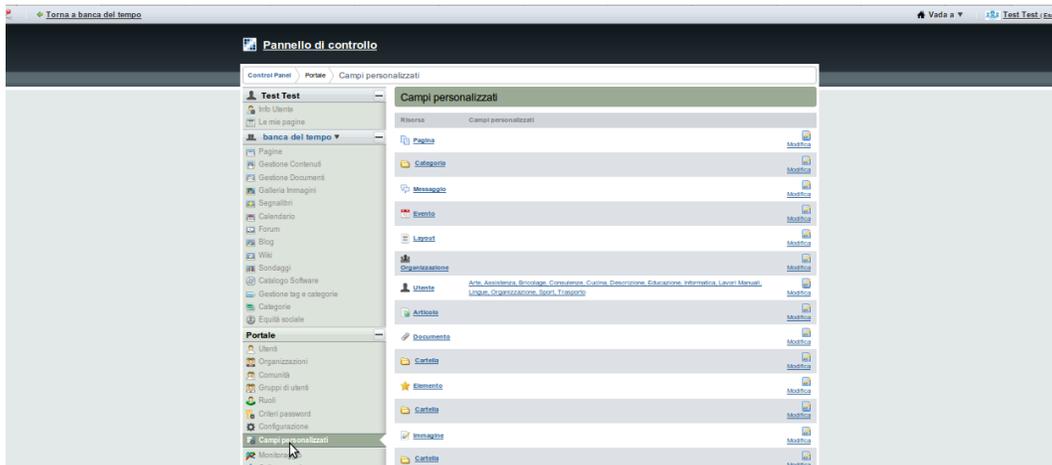


Figura 5.3.1.: Liferay, gestione campi personalizzati

La figura mostra le entità per le quali è possibile definire dei campi personalizzati, all' entità 'Utente' sono stati aggiunti i campi che rappresentano gli interessi di questi ultimi, nonchè le categorie delle quali gli annunci possono far parte.

Tali campi sono stati creati di tipo Boolean, molto importante definire i permessi per gli attributi appena creati, senza una corretta impostazione non sarà infatti possibile, per gli utenti, visualizzare tali campi.

5. Implementazione e integrazione con il portale

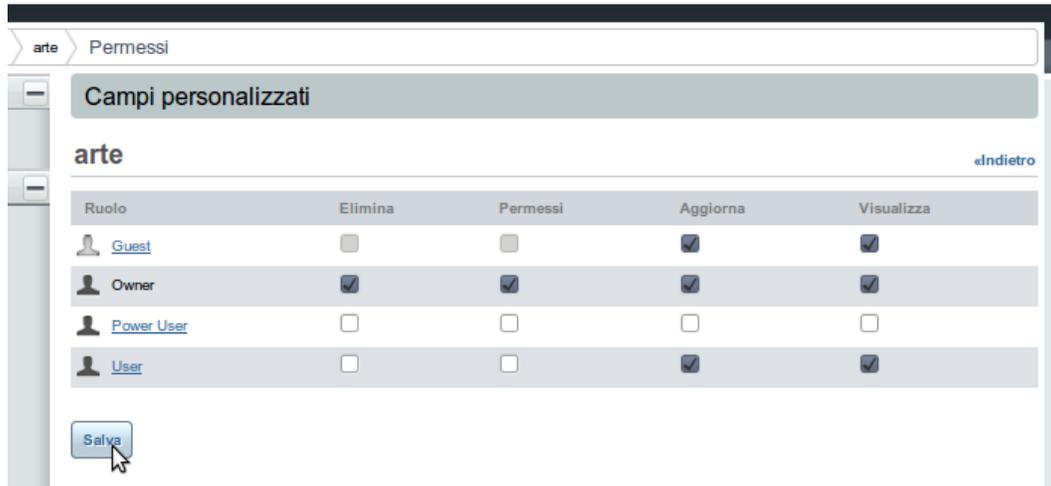


Figura 5.3.2.: Particolare Liferay, gestione permessi

Dopo aver concesso all' utente loggato, come mostra la figura, la possibilità di visualizzare ed aggiornare i dati, è possibile utilizzarli nella pagina di creazione account e nelle classi java del progetto. Di seguito la porzione di codice mostra come recuperare i valori da questa tipologia di campo.

```
import com.liferay.portal.model.User;
2 import com.liferay.portal.service.ClassNameLocalServiceUtil;
import com.liferay.portal.service.UserLocalServiceUtil;
4 import com.liferay.portal.util.PortalUtil;
import com.liferay.portlet.expando.model.ExpandoTableConstants;
6 import com.liferay.portlet.expando.model.ExpandoValue;
import com.liferay.portlet.expando.service.ExpandoValueLocalServiceUtil;
8
long companyId = PortalUtil.getDefaultCompanyId();
10 long classNameId = ClassNameLocalServiceUtil.getClassNameId(User.class);
String tableName = ExpandoTableConstants.DEFAULT_TABLE_NAME;
12 ExpandoValue text = null;
//recupero descrizione
14 text = ExpandoValueLocalServiceUtil.getValue(companyId, classNameId,
tableName, "descrizione", elenco.get(i).getId());
```

Listing 5.2: Codice per la connessione al database

5. Implementazione e integrazione con il portale

La porzione di codice estrapolata dalle portlet “infoUtente” , recupera dalla tabella 'expando table' l' attributo custom 'descrizione'. Il metodo getValue() è soltanto uno tra i molti metodi che l' interfaccia 'ExpandoValueLocalServiceUtil' mette a disposizione dello sviluppatore.

5. Implementazione e integrazione con il portale

5.3.4. Schema relazionale

Di seguito lo schema relazionale che include le tabelle create mediante il 'Service builder' e la tabella 'user_' propria del portale liferay.

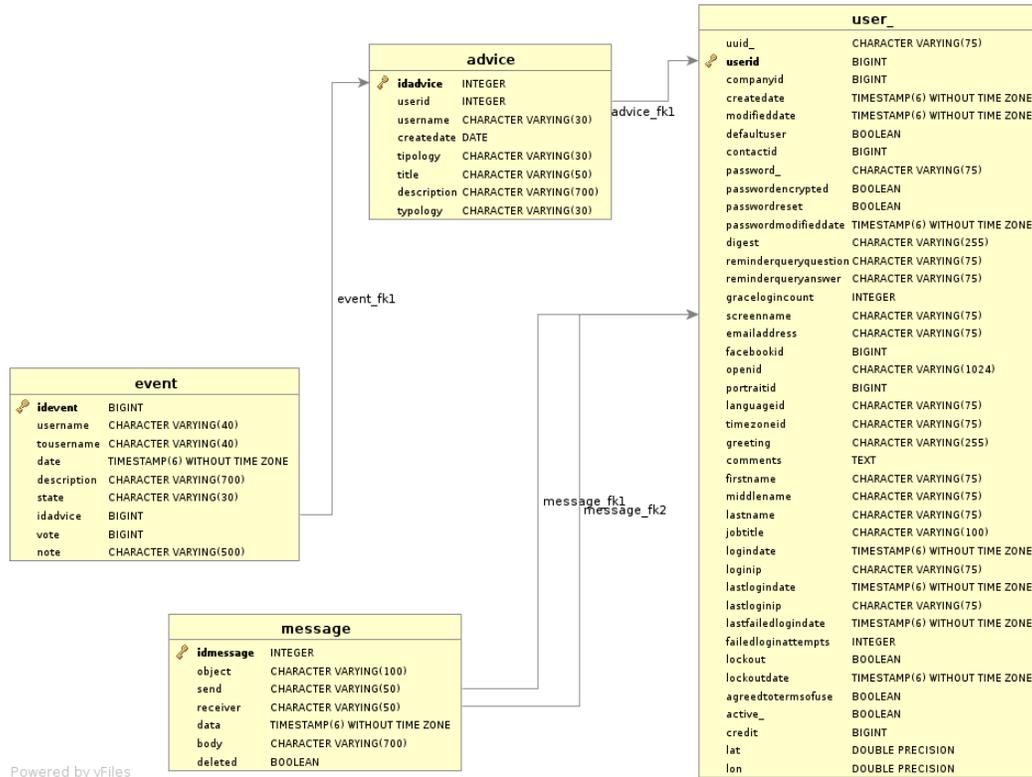


Figura 5.3.3.: Schema relazionale base di dati

Come risulta dallo schema relazionale adesso la tabella “user_” contiene molti più campi rispetto la progettazione prima eseguita. Adesso è possibile conoscere la data di creazione dell’ account, l’ ultimo accesso effettuato, e svariate altre informazioni relative agli utenti registrati nella comunità.

5.4. Vaadin e le portlet

Come prima esposto con l' utilizzo della libreria Vaadin è possibile sviluppare progetti costituiti da moduli altamente scomponibili ognuno dei quali implementa una ben precisa funzionalità.

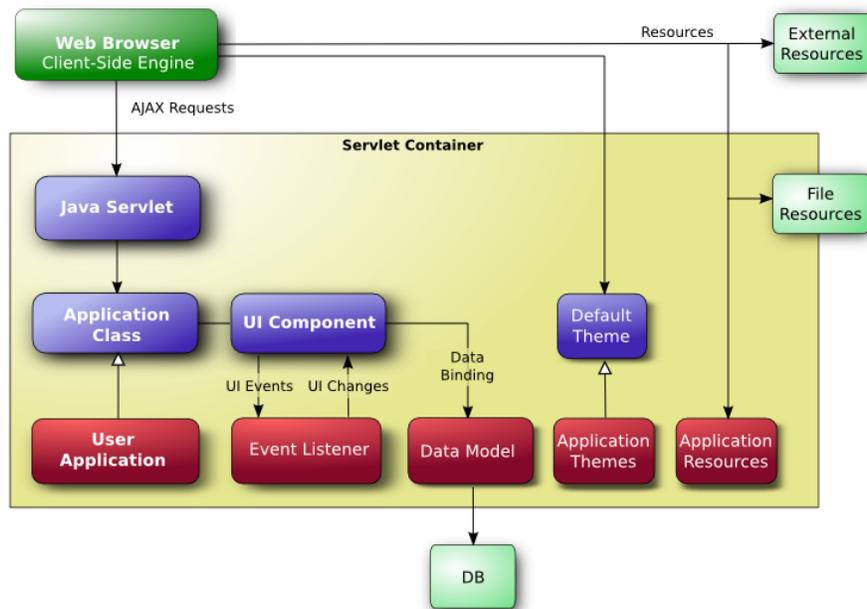


Figura 5.4.1.: Architettura vaadin

Come si evince dalla figura un' intera applicazione sviluppata in Vaadin risiede in un servlet container ed è gestita come fosse una servlet Java. Nel creare un' applicazione lo sviluppatore può utilizzare l' architettura che la libreria mette a disposizione per ottenere applicazioni modulari e altamente scomponibili. Un' applicazione Vaadin è composta, secondo questa architettura, da una classe principale (Application Class) formata da multipli componenti, che si occupano di gestire gli eventi e la sorgente di dati. Prima di essere mostrata a video, all' interfaccia viene aggregata la grafica, i temi e le risorse esterne, che compongono l' applicazione intera .

La figura 5.4.2 mostra le classi e le interfacce con le rispettive gerarchie di ereditarietà.

5. Implementazione e integrazione con il portale

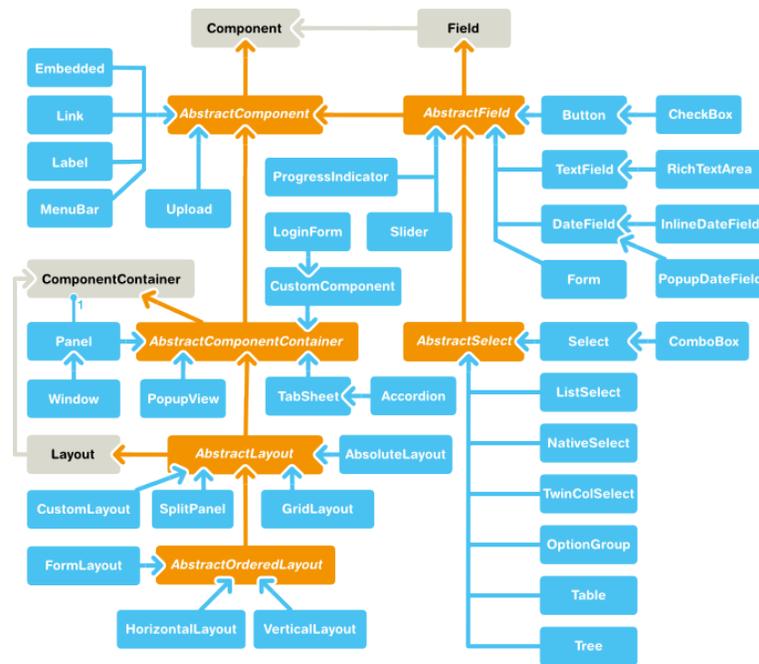


Figura 5.4.2.: Architettura componenti Vaadin

5.4.1. Binding dei dati con Item e Container

Una tra le più utili caratteristiche che vaadin mette a disposizione degli sviluppatori è la possibilità di creare dei componenti direttamente da una sorgente di dati.

Il package `com.vaadin.data` offre allo sviluppatore classi e interfacce, nonché metodi di validazione e controllo sui dati che permettono allo sviluppatore di eseguire operazioni basilari e non, quali ad esempio la creazione di form, la costruzione di tabelle o alberi selezionabili, con un numero esiguo di righe di codice. Esula dagli obiettivi di questo documento una dettagliata argomentazione di tale caratteristica, il codice che crea un Form, comprensivo di validazione sui dati, sarà però d'obbligo per la finalità di questa tesi, e riuscirà bene a definire questa importante caratteristica della libreria utilizzata per questo progetto.

```
1 public class AdviceFormLayout extends Form {
2     public AdviceFormLayout(BeanItem<Advice> adviceItem) {
3         // Creare il layout (3x3 GridLayout)
4         GridLayout gridLayout = new GridLayout(3, 3);
5         // Setta il layout del Form
```

5. Implementazione e integrazione con il portale

```

    setLayout(gridLayout);
7 // Non permessi valori non validi nel DataModel
    setInvalidCommitted(false);
9 // FieldFactory per personalizzare i fields e aggiungere i validatori
    setFormFieldFactory(new AdviceFieldFactory());
11 // binding con l' oggetto passato
    setItemDataSource(adviceItem);
13 // Determina quali properties passare e in quale ordine
    setVisibleItemProperties(Arrays.asList(new String[] { "title",
15 "tipo", "tipology", "date", "description" })); }

//metodo aggiunta field al Form
17 @Override
protected void attachField(Object propertyId, Field field) {
19     if (propertyId.equals("title")) {
        gridLayout.addComponent(field, 0, 0);
21     } else if (propertyId.equals("tipo")) {
        gridLayout.addComponent(field, 2, 0);
23     } else if (propertyId.equals("tipology")) {
        gridLayout.addComponent(field, 0, 2);
25     } else if (propertyId.equals("date")) {
        gridLayout.addComponent(field, 0, 1);
27     } else if (propertyId.equals("description")) {
        gridLayout.addComponent(field, 2, 1, 2, 2); }
29 }
}
```

Listing 5.3: Codice creazione Form, Classe AdviceFormLayout

La classe “AdviceFormLayout” crea un componente griglia (riga 5), e con il metodo `attachField()` vi aggiunge i campi per mezzo dell’ identificativo ‘nome’ (attributo della classe “Advice” che deve implementare i getter e setter di ogni attributo). Le righe 7-11 settano le proprietà del Form, con la riga 13 si dichiara di voler utilizzare l’ oggetto ‘adviceItem’ come sorgente di dati per il Form, mentre la riga successiva esprime gli attributi che verranno visualizzati. Con la riga 11 viene invece richiamata la classe che permette di personalizzare i campi visualizzati.

Di seguito la classe per la personalizzazione del Form.

```

class AdviceFieldFactory extends DefaultFieldFactory {
2 public AdviceFieldFactory() {
    //creo nativeBox categoria
4     for (int i = 0; i < tipologyList.length; i++) {
```

5. Implementazione e integrazione con il portale

```
        tipologySelect.addItem(tipologyList[i]);
    }
6 //creo checkBox tipologia
    typeSelect.setNullSelectionAllowed(false);
8 // user can not 'unselect'
    typeSelect.setValue("richiesta");
10 // i cambiamenti vengono subito
    typeSelect.setImmediate(true);
12 // send the change to the server at once }
    @Override
14 public Field createField(Item item, Object propertyId, Component uiContext) {
        Field f;
16 // formato i campi
        if ("tipology".equals(propertyId)) {
18             tipologySelect.setRequired(true);
                tipologySelect.setValue(tipologyList[7]);
                tipologySelect.setRequiredError("Seleziona categoria annuncio");
                tipologySelect.setWidth(COMMON_FIELD_WIDTH);
22             return tipologySelect;
        } else if ("tipo".equals(propertyId)) {
24             typeSelect.setRequired(true);
                typeSelect.setRequiredError("Inserisci tipologia annuncio");
                typeSelect.setWidth(COMMON_FIELD_WIDTH);
26             return typeSelect;
        } else {
28 // Use the super class to create a suitable field base on the
        f = super.createField(item, propertyId, uiContext); }
30         if ("title".equals(propertyId)) {
            TextField tf = (TextField) f;
32             tf.setCaption("Titolo");
                tf.setRequired(true);
                tf.setRequiredError("Inserisci un titolo");
                tf.setWidth(COMMON_FIELD_WIDTH);
34             tf.addValidator(new StringLengthValidator("Titolo tra 3-50 caratteri", 3, 50, false));
36         } else if ("description".equals(propertyId)) {
            TextField tf = (TextField) f;
40             tf.setCaption("Descrizione");
                tf.setRows(3);
                tf.setColumns(25);
                tf.setRequired(true);
                tf.setRequiredError("Inserisci breve descrizione");
                tf.setWidth(COMMON_FIELD_WIDTH);
                tf.addValidator(new StringLengthValidator("
42                 Descrizione tra 3-500 characters", 3, 500, false));
44             } else if ("date".equals(propertyId)) {
```

5. Implementazione e integrazione con il portale

```
50         DateField tf = (DateField) f;    tf.setCaption("Data");
51         tf.setRequired(true);
52         tf.setRequiredError("Inserisci la data");
53         tf.setWidth(COMMON_FIELD_WIDTH);
54         tf.setDateFormat("dd-MM-yyyy"); }
55     return f; }
56 }
57 }
```

Listing 5.4: Codice creazione Form, Classe AdviceFieldFactory

Questa classe Java estende la classe “DefaultFieldFactory”, attraverso essa è possibile utilizzare i metodi messi a disposizione da Vaadin per la personalizzazione dei Form. L’oggetto “propertyId”, come prima, identifica univocamente l’attributo dell’oggetto per cui deve essere creato il Form, quindi per ogni campo è possibile aggiungere validatori, comprensivi di messaggio di errore, e impostazioni generali, come la dimensione della text area, o la formattazione della data.

Con questo codice è possibile esprimere il concetto di “Data binding” utilizzato da Vaadin. Quando vogliamo popolare un Form, come qualsiasi altro componente (tabelle, alberi, etc.), possiamo servirci di un “container”; un “container” permette di mappare gli oggetti presenti in una base di dati, o in qualsiasi altra sorgente di dati, con gli oggetti utilizzati da Vaadin per la costituzione dei suoi componenti, gli “Items”. I passi necessari a tale scopo sono:

- Istanziare il container.
- Aggiungere oggetti al container.
- Assegnare il container al componente come sorgente di dati.

Per riferirci agli oggetti contenuti nel container si utilizza una delle “propertyes” dell’Item, nel nostro caso utilizziamo la property “propertyId”, che corrisponde al nome dell’attributo delle nostre classi Java.

Grazie a questa architettura è possibile gestire velocemente gli eventi e creare componenti altamente riutilizzabili; ogni qualvolta si avesse bisogno di tale Form nell’applicazione basta richiamare infatti il componente prima creato specificando quale oggetto e quale sorgente di dati utilizzare.

5. Implementazione e integrazione con il portale

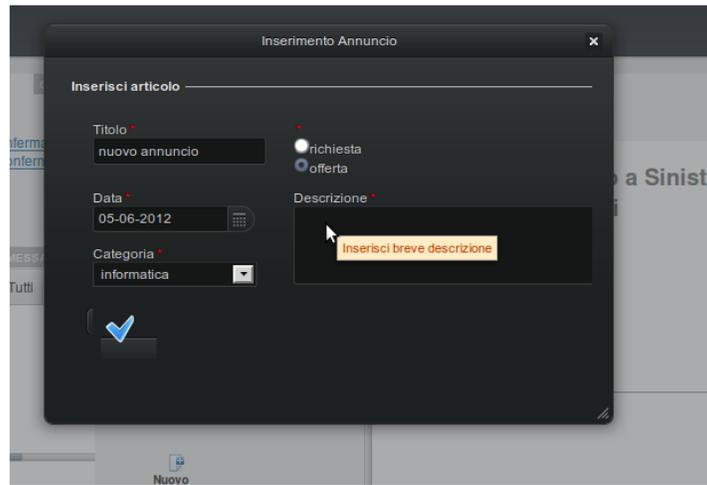


Figura 5.4.3.: Portlet , particolare inserimento annuncio

In questo caso passando l'istanza di un oggetto appena creato, otteniamo un Form di inserimento di un nuovo annuncio e lo aggiungiamo alla finestra principale dell'applicazione.

```
1 Window mainWindow = new Window();  
  //obj Advice  
3 Advice advice = objAdvice;  
  //item Advice  
5 BeanItem<Advice> adviceItem = new BeanItem<Advice>(advice);  
  //Form  
7 final Form adviceForm = new AdviceFormLayout(adviceItem);  
  //aggiungo Form alla finestra  
9 mainWindow.addComponent(adviceForm);
```

Listing 5.5: Codice creazione Form, Classe AdviceFieldFactory

Abbiamo così introdotto il concetto di 'componente'. La finestra di inserimento di un annuncio è, seguendo l'impostazione della libreria Vaadin, un componente utilizzabile in qualsiasi parte della nostra applicazione.

5. Implementazione e integrazione con il portale

5.4.2. Calendar Add-ons

La libreria Calendar, ottenibile nella sezione add-ons del sito ufficiale di Vaadin, permette la visualizzazione e la gestione di eventi offrendo classi e interfacce utili alla rappresentazione di eventi mediante finestre di controllo caratterizzate da una chiara e semplice interazione.

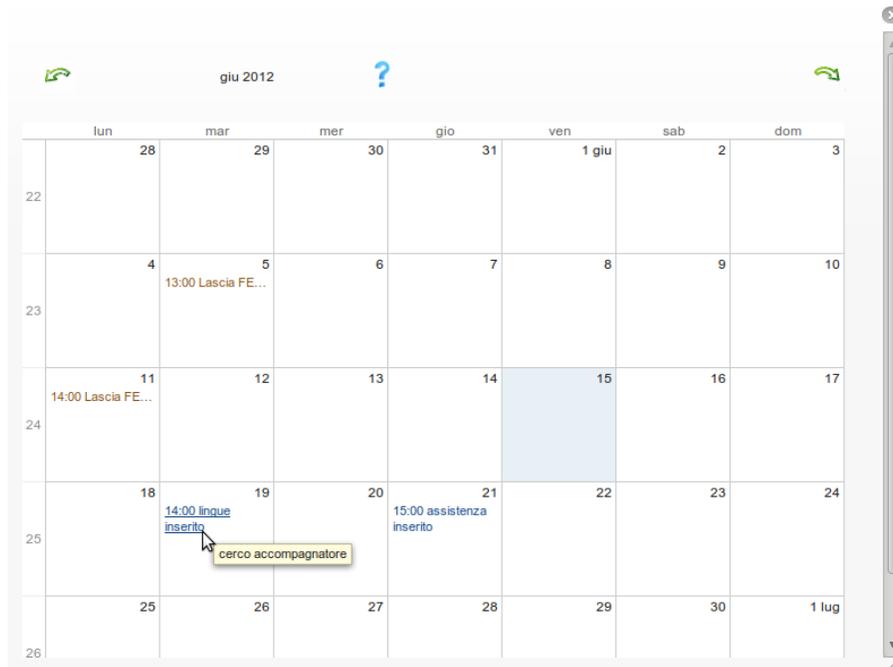


Figura 5.4.4.: Portlet , particolare calendario gestione eventi

La figura mostra la vista mensile del calendario personale di un utente loggato. Aprendo la finestra contenente il calendario, all'utente vengono mostrati gli eventi per i quali ha inserito, o ha ricevuto l'inserimento di un interesse.

In appendice(A1) la Classe CalendarWindow(che estende la classe Component), crea il componente calendario mostrato in figura.

Gli eventi che compongono il calendario sono rappresentati dalla classe CalendarEvent, classe che contiene tutti gli attributi della tabella "event" e implementa i getter ed i setter di tali attributi; per l'inserimento e la modifica di tali eventi viene implementato un Form con la modalità prima descritta.

A. Codice

A.1. Componente calendario

```
1 public class CalendarWindow extends CustomComponent {
3     //lista contenente objs eventi
4     private List<CalendarEvent> elenco;
5
6 public CalendarWindow(Window main){
7     //finestra principale
8     Window mainWindow = main;
9     mywindow = new Window();
10    mainWindow.addWindow(mywindow);
11    //aggiungo calendario alla finestra
12    mywindow.addComponent(calendarComponent);
13    initCalendar();
14    addInitialEvents();
15 }
16
17 //inizializzazione calendario
18 public void initCalendar() {
19     BasicEventProvider dataSource = new BasicEventProvider();
20     calendarComponent = new Calendar(dataSource);
21     calendarComponent.setVisibleHoursOfDay(8, 20);
22     calendarComponent.setLocale(getLocale());
23     calendarComponent.setImmediate(true);
24     calendar = new GregorianCalendar(getLocale());
25     calendar.setTime(getToday());
26     calendar.add(GregorianCalendar.DAY_OF_MONTH, 1);
27     calendar.add(GregorianCalendar.MONTH, 1);
28     calendar.add(GregorianCalendar.DATE, -1);
29     calendarComponent.setStartDate(calendar.getTime());
30     calendarComponent.setEndDate(calendar.getTime());
31 }
32 //aggiunta eventi al calendario
33 public void addInitialEvents() {
34     Date start = elenco.get(i).getStart();
```

A. Codice

```
35         Date end = elenco.get(i).getEnd();
36         CalendarEvent event = getNewEvent("DA LISTA", start, end);
37         String state = elenco.get(i).getState();
38         event.setCaption(elenco.get(i).getCaption() + "<br/>"
+ elenco.get(i).getState());
39         event.setDescription(elenco.get(i).getDescription());
40         event.setState(elenco.get(i).getState());
41         dataSource.addEvent(event);
42         calendar.setTime(originalDate);
43     }
}
```

Listing A.1: Codice creazione calendario, Classe CalendarWindow

B. Regolamento Banca del tempo

ART.1) *L'iscrizione è considerata perfezionata solo dopo l'accettazione delle seguenti regole.*

ART.2) *La Banca del tempo ha come scopo promuovere iniziative atte a favorire lo scambio di attività e di servizi tra persone (come singoli e/o come nuclei familiari), senza alcuna intermediazione di carattere monetario ed aventi, pertanto, come unità di quantificazione il tempo impiegato, e questo al fine di valorizzare i rapporti umani e solidali. Tali prestazioni comunque non dovranno mai poter essere configurate e neppure assimilate a rapporti di lavoro autonomo e subordinato.*

ART.3) *Gli scambi avvengono a partire dal bisogno del singolo partecipante. L'interessato ad una determinata prestazione può contattare direttamente l'offerente e attivare lo scambio; L'interessato ad una determinata prestazione può contattare direttamente il richiedente e attivare lo scambio; A chi offre la prestazione, al termine della stessa, verranno aggiunte le ore pari al lavoro effettuato una volta che chi l'ha richiesta avrà convalidato lo scambio. L'estratto conto di ogni aderente può essere sempre consultabile nelle proprie "pagine riservate". Non è prevista alcuna transizione in denaro e/o valori nello scambio di prestazioni fra i partecipanti. E' ammessa la circolazione di denaro unicamente per l'eventuale rimborso, previamente concordato, delle spese realmente sostenute. Un'ora di tempo impiegato è sempre un'ora, a prescindere dalla prestazione offerta. I rapporti di credito e debito di ogni singolo partecipante sono attuati unicamente con la BdT e non direttamente tra gli aderenti.*

ART.4) *Ogni socio ha diritto ad un credito iniziale di 600 minuti (10 ore). Non ci sono scadenze, rate né interessi passivi, legati a questo credito, per questo si tratta di un credito libero. Il saldo del conto di ogni socio quindi potrà essere talvolta in attivo, talvolta in passivo, senza che il socio debba preoccuparsi di ciò, ma ci si aspetta da ogni socio un movimento sul proprio conto, sia in ingresso che in uscita.*

ART.5) *I soci, all'atto della domanda, indicano un elenco delle proprie capacità e disponibilità. Il socio che ha dato la propria disponibilità è libero però di accettare o di esimersi dalla effettuazione delle singole richieste; in ogni momento può modificare la lista delle proprie disponibilità direttamente nel proprio account, nella propria "pagina riservata".*

ART.6) *La banca del tempo si basa, su un rapporto di fiducia, sul senso di responsabilità e di appartenenza dei soci; ogni scambio è da considerarsi come "obbligazione naturale", si invitano dunque i soci ad essere onesti sulle proprie capacità onde evitare discordie e danni a persone o cose.*

ART.7) *La Banca del tempo si avvale di alcuni strumenti che sono:*

Elenco delle prestazioni offerte: risulta utile per proporre servizi ; Elenco delle prestazioni richieste: serve ad individuare il tipo di prestazione di cui si ha bisogno e il riferimento della persona in grado di offrirla; Elenco dei partecipanti, con nomi cognomi degli associati; Estratto conto periodico, sintesi della contabilità degli scambi effettuati;

B. Regolamento Banca del tempo

Gli strumenti aggiornati sono sempre disponibili all' interno del portale previo accesso autenticato con le credenziali ottenute in fase di iscrizione.

ART. 8) *L'unità di misura delle prestazioni è unicamente il tempo, calcolato in ore.*

Sono esclusi ulteriori frazionamenti. Rimane valida tra i partecipanti la contrattazione per un arrotondamento in eccesso o per difetto.

L'eventuale tempo impiegato negli spostamenti effettuati per erogare la prestazione può essere computato previo accordo fra i partecipanti allo scambio.

ART. 9) *In caso di disaccordo tra i partecipanti allo scambio, l'amministratore è chiamato a pronunciarsi in merito al contenzioso. Per evitare tali controversie, si consiglia agli associati che si accingono a scambiare di definire preventivamente quanto potranno essere le ore impegnate e gli eventuali costi (rimborsi spese) che potranno manifestarsi.*

ART. 10) *Il fondo ore è una sorta di capitale sociale gestito dall'assemblea degli associati. Può essere utilizzato per:*

Coprire le ore di attività di gestione amministrativa ed organizzativa della BdT; Aiutare partecipanti che si trovino in forte debito ed impossibilitati ad effettuare scambi; - Per destinazioni esterne.

ART. 11) *Le modifiche al regolamento vengono definite nell'ambito dell'assemblea dei soci.*

C. Leggi in vigore

Legge 7 Dicembre 2000, n°383. "Disciplina delle associazioni di promozione sociale".

CAPO I DISPOSIZIONI GENERALI

Art.1 (Finalità e oggetto della legge). La Repubblica riconosce il valore sociale dell'associazione liberamente costituito e delle sue attività promuovendone lo sviluppo.

Art.2 (Associazioni di promozione sociale). Sono considerate associazioni di promozione sociale le associazioni riconosciute e non riconosciute, i movimenti, i gruppi o federazioni costituiti al fine di svolgere attività di utilità sociale a favore di associati o terzi, senza finalità di lucro. I partiti politici, le organizzazioni sindacali, le associazioni dei datori di lavoro, le associazioni professionali e di categoria vengono pertanto esclusi. Le Associazioni di promozione sociale devono redigere uno statuto.

Art.3 (Atto costitutivo e statuto). Le associazioni di promozione sociale si costituiscono con atto scritto. Nello statuto devono essere indicati: -l'oggetto sociale; -l'attribuzione della rappresentanza legale dell'associazione; -l'assenza di fini di lucro; -l'obbligo di reinvestire avanzi di gestione in attività istituzionali statutariamente previste; -l'elettività delle cariche associative; -i criteri per l'ammissione e l'esclusione degli associati; -l'obbligo di rendiconti economici-finanziari; -le modalità di scioglimento dell'associazione; -l'obbligo di devoluzione del patrimonio residuo in caso di scioglimento, cessazione o estinzione a fini di utilità sociale.

Art.4 (Risorse economiche). Le associazioni di promozione sociale traggono le risorse economiche per il loro funzionamento da:

-quote, contributi, ereditarietà e donazioni degli associati.

-Contributi dell'Unione Europea e di organismi internazionali.

-Entrate derivanti da iniziative promozionali. Le associazioni devono tenere per tre anni tutta la documentazione apportata in appositi registri a carattere nazionale, regionale e provinciale e così accedere a crediti agevolati o ricevere donazioni o lasciti testamentari.

C. Leggi in vigore

Art.11 (Istituzione e composizione dell'Osservatorio nazionale). L'Osservatorio nazionale dell'associazionismo, denominato "Osservatorio" è preseduto dal Ministro per la solidarietà sociale ed è composto da 26 membri. Dura in carica tre anni ed i componenti possono essere nominati per più di due mandati.

Art.12 (Funzionamento e attribuzioni). L'Osservatorio che ha sede presso il Dipartimento per gli affari sociali, adotta entro 60 giorni un apposito regolamento approvato con decreto del Presidente del Consiglio dei ministri

Legge 23 agosto 1988 n.400.

L'Osservatorio ha competenza di: promozioni di studi e ricerche sull'associazionismo in Italia e all'estero; sostegno delle iniziative di formazione e di aggiornamento; approvazioni di progetti sperimentali; promozioni di scambi di conoscenza e forme di collaborazione fra le associazioni di promozione sociale italiane e straniere. Per l'attuazione del presente articolo nel 2001 è stata autorizzata la spesa massima di lire 1.490 milioni annue.

Art.14 (Osservatori regionali). Le regioni istituiscono osservatori regionali con funzioni e modalità da stabilire con la legge regionale di cui all'art. 8 comma 2. In particolare viene incentivata la costituzione di Banche del Tempo (Titolo1, capo1 art. 43)

Legge N.01 dell'8 Gennaio 2004.

Art.43 (Tempi di cura, tempi di lavoro e tempi delle città) La Regione promuove e incentiva le iniziative di riorganizzazione dei servizi pubblici e privati convenzionati, al coordinamento degli orari e al risparmio di tempo per le attività familiari.

Legge N.07 del 16 Febbraio 2006

determina i criteri e le modalità con cui la Regione riconosce il valore dell'associazionismo favorendone lo sviluppo. Inoltre istituisce il registro regionale delle associazioni, l'Osservatorio regionale e disciplina i rapporti tra le istituzioni pubbliche e le associazioni.

Art.2 (Associazioni di promozione sociale). Sono considerate le associazioni di persone e di enti riconosciute e non se costituite al fine di svolgere attività sociale a favore di terzi o di associati, senza finalità di lucro e con lo scopo di recare benefici alla collettività.

C. Leggi in vigore

Art.3 (Atto costitutivo e statuto delle associazioni di promozione sociale). Le associazioni di promozione sociale si costituiscono con atto scritto indicandone la sede legale. Nello statuto devono essere previsti:

- la denominazione.
- l'oggetto sociale.
- l'attribuzione della rappresentanza legale dell'associazione.
- l'assenza di fini di lucro e la previsione che eventuali proventi delle attività non possono essere divisi fra gli associati.
- l'obbligo di reinvestire avanzi di gestione a favore di attività istituzionali statuarimente previste.
- uguaglianza di diritti di tutti gli associati con la previsione dell'elettività delle cariche associative.
- i criteri per l'ammissione e l'esclusione degli associati ed i loro diritti e obblighi.
- l'obbligo di redazione di rendiconti economici-finanziari e l'approvazione degli stessi.
- le modalità di scioglimento dell'associazione.
- l'obbligo di devoluzione del patrimonio residuo in caso di scioglimento, cessazione o estinzione sociale.

Art.4 (Prestazioni degli associati). Le associazioni possono rimborsare le spese effettivamente sostenute per l'attività prestata entro limiti prestabiliti. In caso di particolare necessità si possono assumere lavoratori dipendenti o avvalersi di prestazioni di lavoro autonomo anche tra i propri associati. Per poter effettuare le attività istituzionali, i lavoratori che fanno parte delle associazioni iscritte nel registro di cui all'art.6 hanno diritto di usufruire di forme di flessibilità dell'orario di lavoro compatibilmente con l'organizzazione aziendale.

Art.5 (Risorse economiche delle associazioni di promozione sociale). Le associazioni di promozione sociale traggono le risorse economiche per il loro funzionamento da:

- quote e contributi degli associati.
- eredità, donazioni e legali.
- contributi dell'Unione Europea e di organismi internazionali.

C. Leggi in vigore

-proventi delle cessioni di beni e servizi degli associati e a terzi.

-altre entrate compatibili con le finalità sociali dell'associazionismo di promozione sociale.

Art.6 (Istituzione del registro regionale delle associazioni di promozione sociale).ore. Il registro regionale della Regione Piemonte si articola in una sezione regionale e in sezioni provinciali. Si possono iscrivere le associazioni che operano da almeno sei mesi e hanno sede operativa in Piemonte. Inoltre, devono essere in possesso dei prerequisiti di cui agli articoli 2 e 3 e l'iscrizione al registro nazionale di cui all'art.7 della L. 383/2000. Tale iscrizione è condizione essenziale per accedere all'assegnazione dei contributi regionali previsti da normative di settore.

Art.7 (Sezione regionale del registro regionale delle associazioni di promozione sociale). Le associazioni di cui all'art. 6, comma 2, che in Piemonte hanno un numero di soci non inferiore a diecimila e che operano almeno in tre province ed iscritte al registro nazionale di cui all'art.7 della L.383/2000 sono iscritte nella sezione regionale del registro.

Art.8 (Sezione provinciale del registro regionale delle associazioni di promozione sociale). Le associazioni di cui all'art.6, comma 2, che rientrano nelle previsioni di cui all'art.7 sono iscritte in base alla località della propria sede legale, nella corrispondente sezione provinciale del registro regionale.

Art.11 (Rapporti con la Regione e con gli enti locali). La Regione, le provincie, i comuni e gli altri enti locali favoriscono lo sviluppo dell'associazionismo di promozione sociale. Altresi' hanno facoltà di stipulare convenzioni co le associazioni di promozione sociale ai sensi dell'art.13. Le Regioni e le provincie per quanto di competenza e limitamente all'accertamento del possesso dei requisiti di iscrizione, possono disporre controlli sulle attività delle associazioni iscritte nelle sezioni del registro regionale di cui all'art. 6.

Art.12 (Riduzione di tributi locali). Gli Enti locali hanno facoltà di deliberare riduzioni o esenzioni sui tributi di propria competenza a favore delle associazioni di promozione sociale iscritte nel registro di cui all'art.6. Art.13 (Convenzioni). La Regione, gli enti locali e gli altri enti pubblici hanno facoltà di stipulare convenzioni con le associazioni di promozione sociale se iscritte almeno da sei mesi nel registro regionale di cui all'art.6 e se hanno presentato un progetto. La copertura assicurativa contro gli infortuni e le malattie e per la copertura assicurativa contro gli infortuni e le malattie e per la responsabilità civile verso terzi di cui all'art.30, comma 3, della L.383/2000 è un elemento essenziale della convenzione.

Art.15 (Sostegno all'associazionismo di promozione sociale). La Regione costituisce un fondo rotativo, gestito dall'Istituto finanziario regionale Finpiemonte S.p.a; finalizzato all'abbattimento dei tassi di interesse sui finanziamenti, per l'acquisto, la costruzione, la ristrutturazione di fabbricati destinati a sede delle attività statuarie delle associazioni. Il finanzi-

C. Leggi in vigore

amento viene erogato su delibera della Giunta regionale e dopo aver sentita la commissione consiliare competente. Inoltre, la Regione concede contributi per le seguenti finalità:

-sostegno alla realizzazione di investimenti in beni materiali e immateriali; per l'acquisto di impianti, macchinari, attrezzature, automezzi, licenze software-

-sostegno alla realizzazione di progetti specifici.

-al potenziamento e alla qualificazione dei servizi erogati.

La Giunta regionale stabilisce le modalità di assegnazione, i criteri di ammissione e valutazione dei contributi di cui al comma 5.

Bibliografia

- [1] Sito nazionale associazioni banche del tempo, <http://www.tempomat.it>
- [2] Frankel N., *Learning Vaadin*, Packt publishing, 2011
- [3] Sito ufficiale Server apache, <http://tomcat.apache.org/>
- [4] Sezov R. jr, *Liferay in Action*, Manning Publications, MEAP Edition, 2010
- [5] Sarin A., *Portlets in Action*, Manning Publications, MEAP Edition, 2010
- [6] Hepper S., *JSR000286 Portlet Specification 2.0 Final Release*, Community development of java technology specifications, 2008
- [7] Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994