



Specifica tecnica per C04/PQS

Versione 3.0

Stato del documento:

Formale ed

Esterno

Sommario :

Questo documento si prefigge di illustrare l'architettura del prodotto relativo al capitolato C04: "Sistema *software* per l'informatizzazione della gestione di qualità a norma ISO 9000:2000 nelle scuole".

Redazione

Nominativo	Ruolo	Data
Eric Miotto	Progettista	28 gennaio 2007
Stefano Gazzola	Progettista	28 gennaio 2007

Lista di Distribuzione

Nominativo	Ruolo
Tullio Vardanega	Committente
Renato Conte	Committente
Lucia Meneghello	Verificatore
Margherita Collicelli	Responsabile
Eric Miotto	Amministratore/Programmatore
Stefano Gazzola	Programmatore
Roberto Pordon	Verificatore
Lorenzo Daniele	Programmatore
Alberto Meneghello	Programmatore

Approvato da:

Versione	Nominativo	Data
3.0	Margherita Collicelli	16 marzo 2007

Registro delle Modifiche:

Versione	Autore	Data
3.0	Margherita Collicelli	16 marzo 2007
Approvazione documento per la Revisione di Qualifica.		
2.3	Lucia Meneghello	16 marzo 2007
Verificato documento.		
2.29	Eric Miotto	16 marzo 2007
Aggiornata Illustrazione 4.		
2.28	Lorenzo Daniele	14 marzo 2007
Aggiunto diagramma di sequenza ed aggiornati altri diagrammi (3.1.5).		
Modifiche varie		
2.27	Lorenzo Daniele	14 marzo 2007
Diagramma di sequenza ed aggiornati altri diagrammi (3.1.5).		
2.26	Lorenzo Daniele	13 marzo 2007
Inserito e steso capitolo sull'architettura della GUI (3.1.5).		
2.26	Lorenzo Daniele	13 marzo 2007
Inserito e steso capitolo sull'architettura della GUI (3.1.5).		
2.25	Eric Miotto	12 marzo 2007
Aggiornata Illustrazione 4.		
2.2	Eric Miotto	12 marzo 2007
Aggiornata sezione 3.2.3 (riviste classi ricerca e spostata loro documentazione all'esterno).		
2.1	Eric Miotto	09 marzo 2007
Correzioni varie. Aggiornata sezione 3.2.3 (spezzata interfaccia WS Didattica, spostata all'esterno documentazione WS). Specificate meglio le tecnologie nella sezione 4..		
2.01	Lucia Meneghello	08 marzo 2007
Aggiornamento frontespizio come da raccomandazioni RPD.		
2.0	Lorenzo Daniele	23 febbraio 2007
Approvazione documento per la Revisione di Progetto Definitiva.		
1.08	Lucia Meneghello	23 febbraio 2007
Completata verifica del documento.		
1.07	Eric Miotto	22 febbraio 2007
Raffinata descrizione relazioni nella descrizione delle classi nella sezione 3.2.3		

Egoless Group

1.06	Lucia Meneghello	22 febbraio 2007
Aggiornata lista di distribuzione. Verifica del documento con relative correzioni.		
1.05	Stefano Gazzola	22 febbraio 2007
Aggiunto riferimento all'utilizzo del facade pattern nel capitolo 2.3		
1.04	Margherita Collicelli	21 febbraio 2007
Finita la conversione da inglese a italiano.		
1.03	Stefano Gazzola	18 febbraio 2007
Cominciata la conversione da inglese ad italiano della parte di interfaccia del web service		
1.02	Stefano Gazzola	18 febbraio 2007
Inserito riferimento al pattern MVC nell'APP-DIDATTICA		
1.01	Eric Miotto	18 febbraio 2007
Rivista sezione 2.2. Aggiornata Illustrazione 1 ed inserita legenda per la sua lettura.		
1.0	Roberto Pordon	02 febbraio 2007
Approvazione documento per la Revisione Preliminare di Progetto		
0.21	Alberto Meneghello, Lorenzo Daniele	01 febbraio 2007
Verifica del documento. Corrette alcune discrepanze sintattiche.		
0.20	Stefano Gazzola	01 febbraio 2007
Integrazione capitolo 3.1.4.		
0.19	Stefano Gazzola	01 febbraio 2007
Controllo e correzione generale.		
0.18	Stefano Gazzola	31 gennaio 2007
Inserita matrice di tracciamento componenti-requisiti, integrazione del formalismo di specifica con la spiegazione della nomenclatura utilizzata. Inserimento descrizione classi e diagrammi delle classi.		
0.17	Eric Miotto	31 gennaio 2007
Inserita prima bozza su WS-Didattica (sezione 3.2). Diviso in due sezioni la descrizione del metodo e del formalismo di specifica (sezioni 2.1 e 2.2).		
0.16	Stefano Gazzola	30 gennaio 2007
Modifiche capitolo 3.2		
0.15	Lucia Meneghello	30 gennaio 2007
Operate correzioni ortografiche e formali al testo. Aggiornamento Lista di distribuzione.		
0.14	Alberto Meneghello, Lorenzo Daniele	29 gennaio 2007
Aggiunti termini al glossario.		

Egoless Group

0.13	Eric Miotto	28 gennaio 2007
Operata ulteriore correzione del testo. Abbozzata le descrizioni dei componenti nella sezione . Aggiunta sezione sulla collaborazione (sezione 1.3)		
0.12	Eric Miotto	28 gennaio 2007
Prima correzione del testo.		
0.11	Stefano Gazzola	28 gennaio 2007
Inserito capitolo 1 (1.1, 1.2) e capitolo 2 (2.1, 2.2)		
0.1	Eric Miotto	29 novembre 2006
Prima bozza del documento		

Indice

1	Introduzione.....	7
1.1	Scopo del documento.....	7
1.2	Scopo del prodotto.....	7
1.3	Collaborazione con Swell Systems.....	7
1.4	Riferimenti.....	7
2	Definizione del prodotto.....	8
2.1	Metodo di specifica.....	8
2.2	Formalismo di specifica.....	8
2.3	Presentazione dell'architettura generale del sistema e identificazione dei componenti architetturali di alto livello.....	8
2.3.1	<i>Studio di fattibilità tecnologica a supporto dell'architettura scelta.....</i>	<i>10</i>
3	Descrizione dei singoli componenti.....	12
3.1	APP Didattica [CSA-01].....	12
3.1.1	<i>Tipo, obiettivo e funzione.....</i>	<i>12</i>
3.1.2	<i>Relazioni d'uso di altre componenti.....</i>	<i>15</i>
3.1.3	<i>Interfacce con e relazioni di uso da altre componenti.....</i>	<i>15</i>
3.1.4	<i>Attività svolte e dati trattati.....</i>	<i>15</i>
3.1.5	<i>Architettura della GUI.....</i>	<i>15</i>
	Model.....	15
	View.....	15
	Controller.....	16
3.2	WS-Didattica [CSW-01].....	18
3.2.1	<i>Tipo, obiettivo e funzione.....</i>	<i>18</i>
3.2.2	<i>Relazioni d'uso di altre componenti.....</i>	<i>18</i>
3.2.3	<i>Interfaccia esposta verso altre componenti.....</i>	<i>18</i>
4	Stime di fattibilità e di bisogno di risorse.....	27
5	Tracciamento della relazione componenti - requisiti.....	27

1 Introduzione

1.1 Scopo del documento

Questo documento si propone di definire l'architettura del nostro sistema software ad un livello di astrazione molto alto al fine di capirne il funzionamento logico. Tale architettura sarà il mezzo attraverso cui cercheremo di raggiungere gli obiettivi di progetto.

Inoltre saranno definiti i componenti costitutivi e la loro organizzazione, individuando le interfacce che permettono ai componenti stessi di interagire fra loro.

1.2 Scopo del prodotto

Per lo scopo del prodotto fare riferimento a [AR].

1.3 Collaborazione con Swell Systems

L'intero prodotto viene sviluppato in collaborazione con l'altro gruppo che ha scelto il capitolato C04, Swell Systems. La collaborazione è dovuta a due motivi:

- il capitolato è tale per cui è auspicabile (e sensata) la produzione di un solo prodotto;
- le risorse e il tempo a disposizione di entrambe i gruppi non permettono di giungere ad un prodotto che soddisfi i requisiti.

Con la collaborazione si spera di costruire una buona infrastruttura che possa essere evoluta in futuro per arrivare alla costruzione di un ottimo sistema informativo che risponda ai requisiti del capitolato.

1.4 Riferimenti

- [AR] Analisi dei Requisiti, Egoless Group
- [G] Glossario, Egoless Group
- [PP] Piano di progetto, Egoless Group
- [NP], Norme di Progetto, Egoless Group
- [DP], Definizione di prodotto, Egoless Group
- [DocIWS], Documentazione interfacce WS
- [DocCRIC], Documentazione classi per la ricerca

2 Definizione del prodotto

2.1 Metodo di specifica

La specifica si svilupperà per gradi di astrazione partendo da un livello più alto in cui si illustra l'architettura suddividendola nelle sue componenti più significative che poi verranno via via smembrate per essere analizzate più dettagliatamente.

Questa specifica sarà integrata con nuove informazioni ogni qualvolta una nuova componente verrà aggiunta, analizzata o approfondita. In particolare in questa prima iterazione verrà preso in considerazione il modulo didattico (CU09). Questa scelta andrà ad influire anche sul lavoro di Swell Systems che darà maggiore priorità alle esigenze di questo modulo.

2.2 Formalismo di specifica

Per la stesura dei diagrammi si seguiranno le norme riportate in [NP].

Nella descrizione dei Web Service si modelleranno le classi che generano il Web Service e non il WSDL (Web Service Description Language).

Ciascun componente individuato nell'architettura sarà identificato da una sigla con la seguente struttura:

$$CSx-y/z$$

dove "CS" sta per "Componente software", la variabile x può assumere il valore W (Web service) oppure A (Applicativo), la variabile y è un numero progressivo espresso con due cifre (01,02,...11,...), la variabile z indica la numerazione delle classi di ciascun componente con un numero progressivo(1,2,3,...).

2.3 Presentazione dell'architettura generale del sistema e identificazione dei componenti architetturali di alto livello

Data l'impossibilità di realizzare l'intero prodotto richiesto, nel tempo pianificato (vedi [PP]), la scelta dell'architettura è stata fatta in modo tale da facilitare la manutenzione e il riuso a lungo termine, in modo che l'investimento fatto possa durare nel tempo.

L'architettura ad alto livello che sarà presentata nel seguito si basa su di un particolare *pattern architetturale* detto "multi-layer pattern" appartenente alla famiglia più generale del "layers pattern". La comunicazione tra i vari livelli avviene attraverso Web Service. L'uso dei Web Service permette di avere riuso e interoperabilità.

La scelta di questo tipo di pattern architetturale nasce dall'esigenza di decomporre e catalogare la struttura dell'applicazione in gruppi di *moduli* che per loro intrinseca natura hanno esigenze e bisogni comuni.

Individuati i moduli si cercherà, quindi, di isolarli e di renderli il più indipendenti possibile attraverso le tecnologie utilizzate.

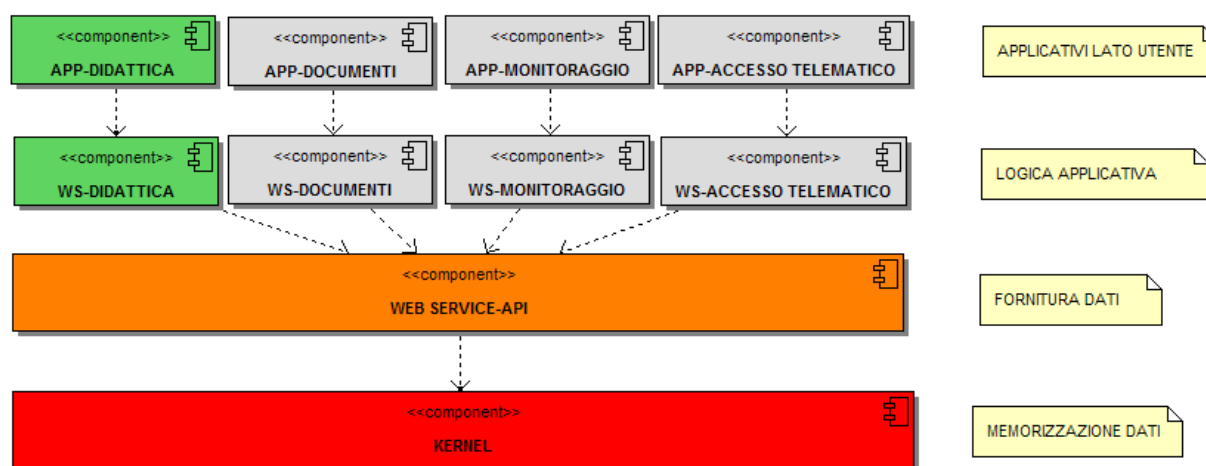


Illustrazione 1: Architettura generale del sistema

Nell'illustrazione 1 è riportata l'architettura individuata. Le componenti sono state colorate in base al gruppo che ne è responsabile:

- le componenti colorate di **verde** sono progettate e realizzate da Egoless Group;
- le componenti colorate di **arancione** sono progettate congiuntamente da Egoless Group e Swell Systems e realizzate da Swell Systems;
- le componenti colorate di **rosso** sono progettate e realizzate da Swell Systems;
- le componenti colorate di **grigio** sono solamente previste ma non progettate e realizzate da nessun dei due gruppi.

Qui di seguito è riportato il tracciamento dei componenti e il loro identificativo:

Componente	Sigla
APP-DIDATTICA	CSA-01
APP-DOCUMENTI	CSA-02
APP-MONITORAGGIO	CSA-03
APP-ACCESSO TELEMATICO	CSA-04
WS-DIDATTICA	CSW-01

Componente	Sigla
WS-DOCUMENTI	CSW-02
WS-MONITORAGGIO	CSW-03
WS-ACCESSO TELEMATICO	CSW-04

L'architettura individuata è schematizzata in Illustrazione 6 ed è formata dalle seguenti componenti:

- APP–Didattica, APP–Documenti, APP–Monitoraggio, APP–Accesso telematico: insieme di applicativi lato *client* e *desktop-oriented*. Permettono all'utente di interagire con il sistema e di ottenere il servizio voluto. Ogni applicativo può essere installato su più macchine differenti. Si è scelto di sviluppare questi moduli come applicazioni desktop per poter realizzare un'interfaccia grafica *user-friendly* e possibilmente molto sofisticata.
- WS–Didattica, WS–Documenti, WS–Monitoraggio, WS–Accesso telematico: insieme di web service con i quali ciascun tipo di applicativo interagisce. L'utilizzo di web service permette di disaccoppiare l'interfaccia utente dalla parte logica permettendo un buon grado di isolamento e di riuso dei servizi. Questo permette di poter modificare, evolvere, sostituire l'implementazione del servizio in modo del tutto indipendente dai livelli superiori e inferiori.
- WEB SERVICE–API: interfaccia che sancisce quali informazioni i web service di livello superiore possono richiedere e sulle quali possono lavorare, indipendentemente da come tali informazioni verranno concretizzate come dati di un database. Questo livello si propone esso stesso a livello superiore come web service per rendere il servizio offerto facilmente adattabile a cambi strutturali e tecnologici (es. permettere un consorzio di scuole per la gestione in comune di certi servizi rivolti ai dati).
- *KERNEL*: insieme formato dal database che memorizza i dati della scuola e dalle funzionalità atte alla sua gestione.

L'utilizzo dei Web service rende evidente la presenza dello structural pattern Facade in quanto viene esposta una interfaccia comune alla quale i consumatori del servizio devono far riferimento per accedere alle funzionalità del sistema sottostante.

2.3.1 Studio di fattibilità tecnologica a supporto dell'architettura scelta

L'architettura presentata precedentemente potrebbe risultare abbastanza utopistica nel caso in cui non venga supportata da uno studio di fattibilità tecnologica che possa

giustificare le scelte fatte. Tali soluzioni e alcune di quelle scartate sono passate attraverso uno studio dell'impatto che potevano avere a livello fisico e prestazionale.

Le necessità dell'ambiente nel quale il nostro sistema andrà ad innestarsi hanno dettato questo tipo di comportamento.

All'interno del sistema scuola infatti è stato individuato il requisito non funzionale RN14.B che impone di utilizzare configurazioni hardware spesso datate o comunque di poter prevedere l'innesto di configurazioni con un basso impatto economico sulle casse della scuola stessa.

Allo stato attuale delle cose inoltre dobbiamo considerare che il nostro sistema potrebbe girare interamente all'interno di un'unica macchina: è inevitabile quindi che dovranno essere presenti nello stesso momento più sistemi software (*DBMS*, applicativi, JVM...) che rivendicheranno il proprio spazio e le proprie esigenze.

D'altro canto, è necessario porre dei requisiti hardware minimi tali da garantire il buon funzionamento del sistema.

Analizzando dall'alto l'architettura proposta incontriamo lo strato degli “applicativi utente” che in base alle nostre previsioni verranno sviluppate in linguaggio Java. Considerato che un IDE completo e molto complesso sviluppato in Java come NetBeans (www.netbeans.org) ha come requisiti di sistema un Pentium 3 con processore 500 Mhz e 512MB di RAM con valore di mercato piuttosto irrisorio, i nostri applicativi ben meno complessi necessiteranno di una configurazione hardware inferiore.

Un livello sotto incontriamo la “logica applicativa” basata su web-service che dovrà gestire accessi di tipo concorrenziale. Partendo dal presupposto che un limitato numero di persone avranno il diritto di accedere alle funzionalità proposte, ogni web-service (e quindi per ogni applicativo a livello superiore) dovrà gestire un numero di accessi contemporanei che nel caso peggiore risulta essere un numero molto basso (dell'ordine delle decine). Nel caso medio tale numero sarà ancora più irrisorio. Con questo possiamo giustificare anche il fatto che nonostante il web-service utilizzi per le comunicazioni XML (quindi operando onerose serializzazioni/deserializzazioni da/verso XML) l'impatto viene tranquillamente assorbito proprio perché le richieste saranno molto diluite nel tempo.

Discorso duale può essere fatto per i *layer* sottostanti con l'unica nota rivolta alla base di dati e al DBMS che verranno utilizzati i quali dovranno gestire una base di dati abbastanza ridotta e quindi l'impatto di un DBMS come MySQL verrà tranquillamente assorbito.

L'unica risorsa che deve essere massimizzata (sempre rispettando i limiti imposti) è la memoria Ram: tra i 512 MB e 1024 MB i costi allo stato attuale sono molto contenuti

(di molto inferiori ai 100 €) quindi un piccolo investimento su questo fronte può essere assorbito nel tempo.

Per quanto riguarda il disco rigido anche qui capienze più che sufficienti (≤ 100 GB) si possono trovare con costi molto inferiori a 100 € anche se sono risorse che una scuola facilmente può recuperare in loco.

3 Descrizione dei singoli componenti

Verranno descritte le componenti legate ai requisiti obbligatori in [AR]. Non verranno descritti i moduli di competenza di Swell Systems.

3.1 APP Didattica [CSA-01]

3.1.1 Tipo, obiettivo e funzione

APP Didattica è l'applicazione desktop usata per gestire i dati di studenti, docenti ed infrastrutture. L'obiettivo principale è quello di fornire un'interfaccia grafica il più completa e semplice possibile.

Di seguito si trovano alcuni *screenshot* che rappresentano la prima bozza di interfaccia grafica la quale ha lo scopo di organizzare le funzionalità al fine di avere un primo contatto con i possibili problemi a livello visivo che potrebbero sorgere.

Al fine di una corretta visualizzazione li abbiamo ruotati.

Gestione della Didattica
File Edit Help

Gestione Studenti | Gestione Personale | Gestione Infrastrutture | Registro Docente | Registro Del Docente

Ricerca Studente

☐ Ricerca per matricola

☐ Ricerca per cognome-nome

☐ Ricerca per classe-sezione

Avvia ricerca

Risultati Ricerca

Nome	Cognome	Matricola	Data di nascita	Classe	Sezione

Ordina per
☐ Nome
☐ Cognome
☐ Sezione
☐ Classe

Carica dati studente

Modifica-Inserimento dati dello studente

Genera

Matricola

Nome

Cognome

Data di nascita

...

...

Salva Annulla Pulisci

Carriera studente

Materia1	Mater...	...	MEDIA	CREDITI	DEBITI

Illustrazione 2: Screenshot della GUI "gestione studenti" dell'applicativo didattica

Questa GUI presenta la schermata "Gestione studenti" che sarà utilizzata principalmente dal personale della segreteria per il trattamento dei dati sensibili dello studente

Gestione della Didattica
File Edit Help

Gestione Studenti | Gestione Personale | Gestione Infrastrutture | **Registro Del Docente**

Ricerca Studente
☐ Ricerca per matricola
☐ Ricerca per cognome-nome
☐ Ricerca per classe-sezione
 Avvia ricerca

Risultati Ricerca

Nome	Cognome	Matricola	Data di nascita	Classe	Sezione

Ordina per
☐ Nome
☐ Cognome
☐ Sezione
☐ Classe

Carica dati studente

Voti Studente

Id_voto	Data	Descrizi...	Tipo	Valore

Data: Tipo: Valore: Descrizione:
 Nuovi Modifica Salva Annulla Pulisci

Assenze studente

Id	Data inizio	Data fine	Giustific...	Note

Id: Genera Data inizio: Data fine: Note:
 Nuova Modifica Salva Annulla Pulisci

Illustrazione 3: Screenshot della parte “registro docente” dell'applicativo didattica

Questa GUI presenta la schermata “Registro docente” che sarà utilizzata principalmente dal docente per la gestione dei voti e delle assenze dello studente.

3.1.2 Relazioni d'uso di altre componenti

Il componente usa WS Didattica per applicare le modifiche che l'utente ha specificato. L'insieme delle funzionalità che tale applicativo utilizzerà sono quelle messe a disposizione dall'interfaccia del Web Service.

3.1.3 Interfacce con e relazioni di uso da altre componenti

Il componente non è usato da nessun'altra componente perché si interfacerà direttamente con l'utente.

3.1.4 Attività svolte e dati trattati

Il componente sarà strutturato internamente in più parti per la produzione dei dati da passare al Web Service sottostante.

Saranno individuate e implementate le parti seguenti:

- una parte di GUI (vedi screenshot) per l'inserimento e la visualizzazione dei dati;
- una parte logica per un controllo preliminare della validità dei dati inseriti e per l'elaborazione dei dati da visualizzare.
- una parte di preparazione dei dati per lo scambio con il Web Service sottostante;

La struttura del componente sarà modellata secondo il pattern architetturale MVC. Tutti i dettagli sono specificati nella DP.

3.1.5 Architettura della GUI

Nel progettare la GUI si è scelto di usare come linea guida il pattern architetturale MVC (Model – View – Controller). Questo permette di separare logicamente il modello dei dati (stato dell'applicazione) dalla loro presentazione (view) e dal controller che si occupa di trattare gli eventi generati dagli utenti e di notificare cambiamenti nel model alla view. Il maggiore vantaggio di questa architettura è che possono essere create viste multiple ed indipendenti che condividano lo stesso model attraverso i rispettivi controller.

Model

Model è il cuore dell'applicazione. Definisce i dati e le operazioni che possono essere eseguiti su essi. Fornisce delle funzioni per l'accesso e l'aggiornamento. Può inoltre avere la responsabilità di notificare ai componenti della View eventuali aggiornamenti verificatisi in seguito a richieste del Controller, al fine di permettere alle View di presentare agli occhi degli utenti dati sempre aggiornati.

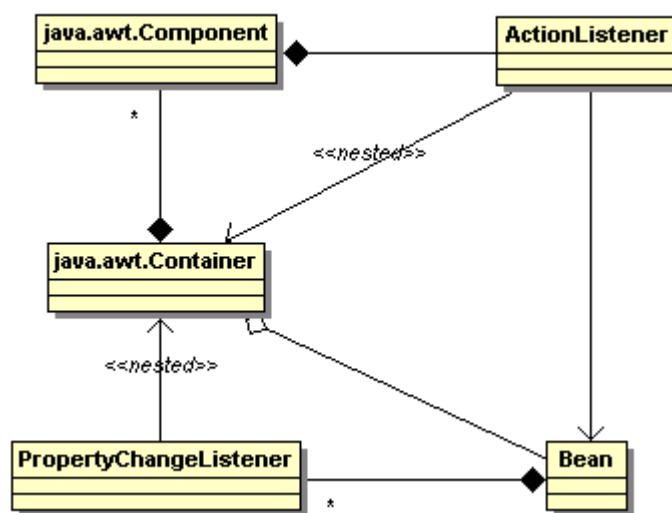
View

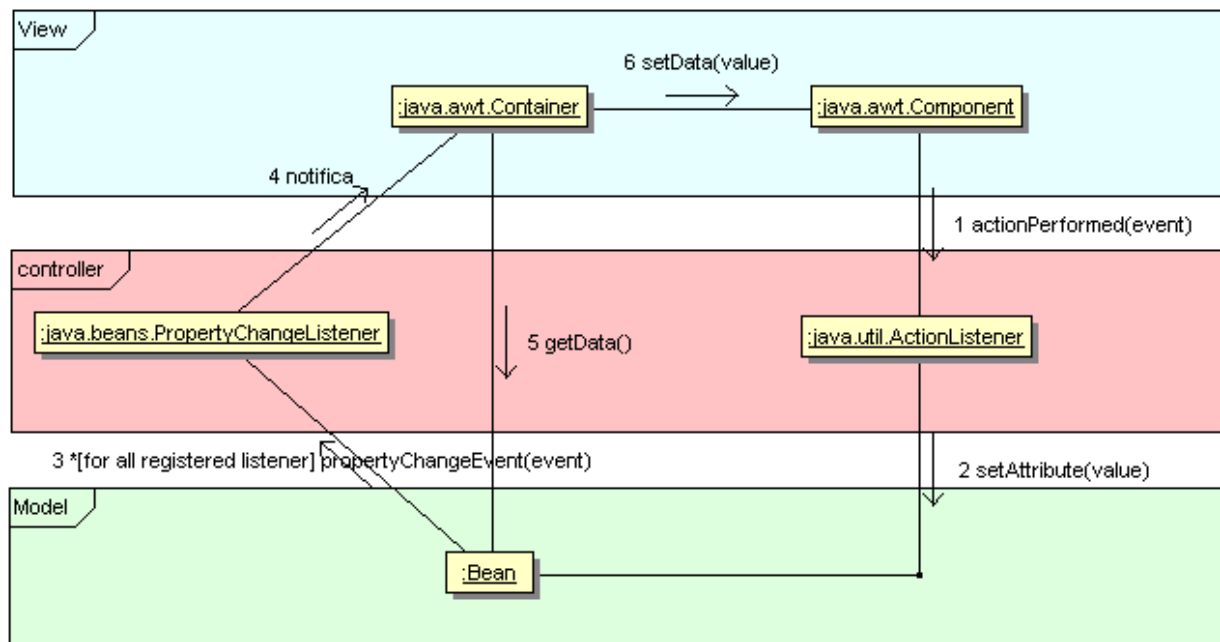
View è l'interfaccia grafica (GUI) con cui l'utente interagisce.

Controller

Il controller ha la responsabilità di trasformare le interazioni dell'utente della View in azioni eseguite dal Model. Ma il Controller non rappresenta un semplice "ponte" tra View e Model. Realizzando la mappatura tra input dell'utente e processi eseguiti dal Model e selezionando la schermate della View richieste, il Controller implementa la logica di controllo dell'applicazione.

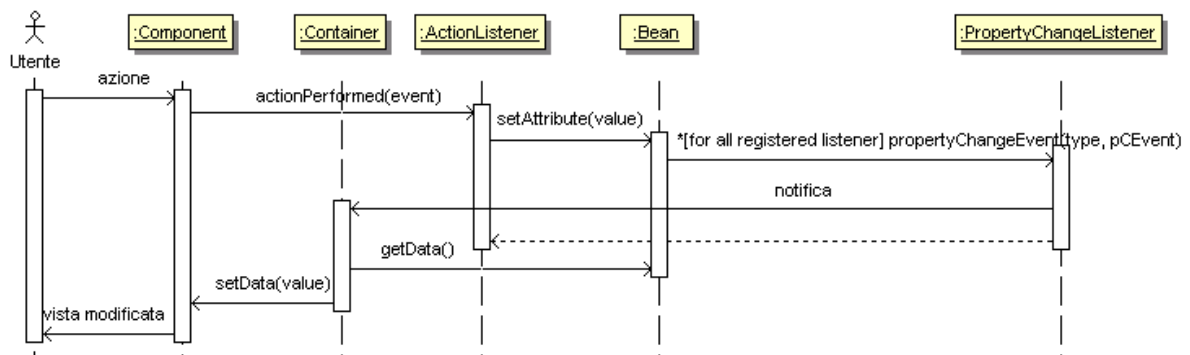
I seguenti schemi delle classi e di comunicazione mostrano come viene da noi implementato tale pattern utilizzando gli strumenti messi a disposizione dalla piattaforma Java.





In realtà le notifiche vengono fatte su tutti gli oggetti che si sono registrati presso il Bean (cioè il Bean chiama il metodo su ognuno dei `PropertyChangeListener` registrati presso di lui) e di conseguenza tali oggetti potrebbero modificare la presentazione di molteplici oggetti indipendenti tra loro e non correlati (in questo modo è possibile creare viste multiple per lo stesso model).

Il seguente diagramma di sequenza rappresenta una vista alternativa al precedente diagramma di collaborazione



3.2 WS-Didattica [CSW-01]

3.2.1 Tipo, obiettivo e funzione

WS Didattica implementa tutta la logica per gestire i dati di docenti e studenti. In particolare:

- permette di gestire gli studenti e i professori;
- per ogni studente permette di gestire i voti e le assenze;
- permette di gestire informazioni di supporto (tipo di voto, tipo di assenza, ...)

3.2.2 Relazioni d'uso di altre componenti

Per svolgere le sue funzioni, WS-Didattica si appoggia a WEB SERVICE-API:

- per recuperare e salvare in modo permanente tutte le informazioni della scuola;
- per avere le strutture dati fondamentali con le quali svolgere il proprio lavoro.

Allo stato attuale di questo documento le strutture dati che sono condivise tra i due livelli possono presentare alcune discrepanze. Questa situazione è causata dal fatto che la definizione delle classi nei due livelli sono ancora molto fluide. Le due parti stanno lavorando per convergere verso una soluzione comune.

3.2.3 Interfaccia esposta verso altre componenti

Il componente espone una serie di Web Service usati da APP-Didattica. Ogni Web Service è responsabile di un aspetto della didattica.

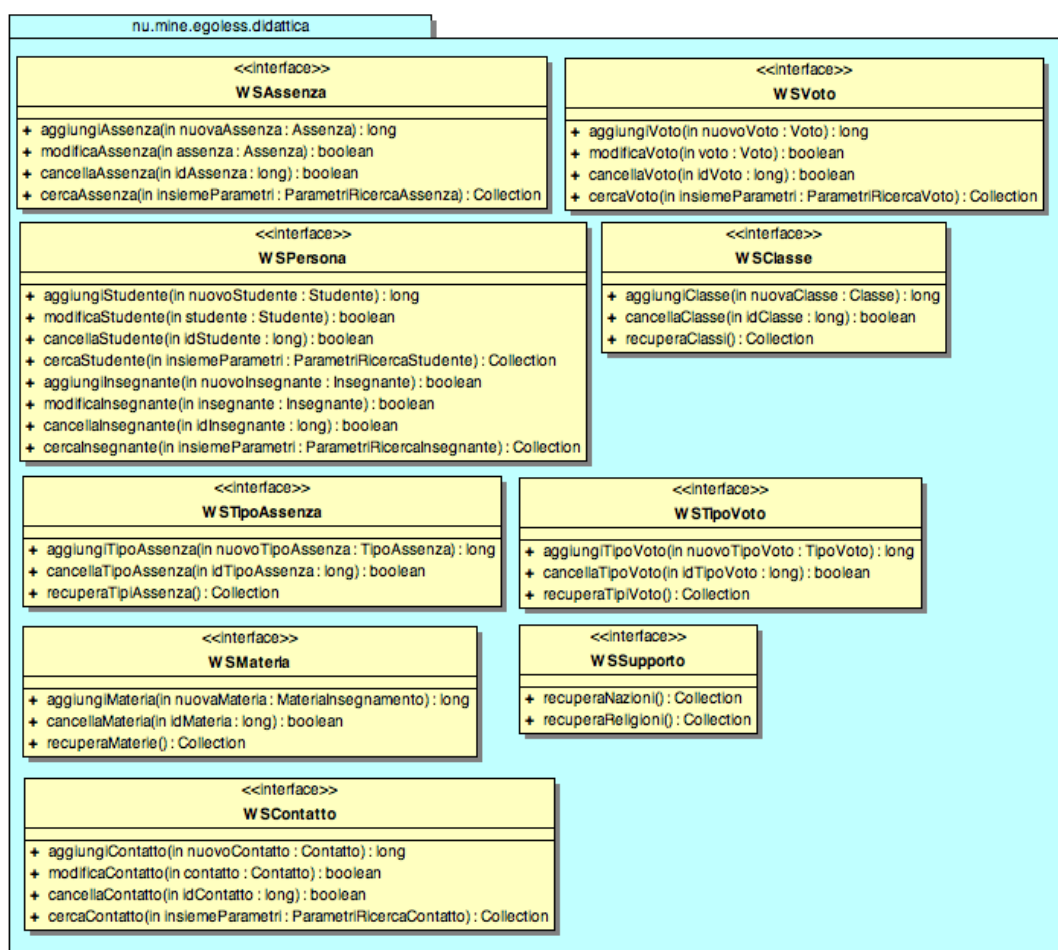


Illustrazione 4: Elenco dei metodi esposti dall'interfaccia del web service

Le classi esposte dal Web Service possono essere divise nelle seguenti categorie:

- classi dati ereditate da WEB SERVICE API. Queste si dividono ulteriormente in
 - **classi semplici:** corrispondono ad informazioni contenute che si mantengono relativamente stabili nel tempo. Per ogni classe ci sono poche istanze;

- **classi complesse:** corrispondono ad informazioni rilevanti per la scuola, che sono molto grandi e che mutano continuamente nel tempo. Per ogni classe ci sono molte istanze.

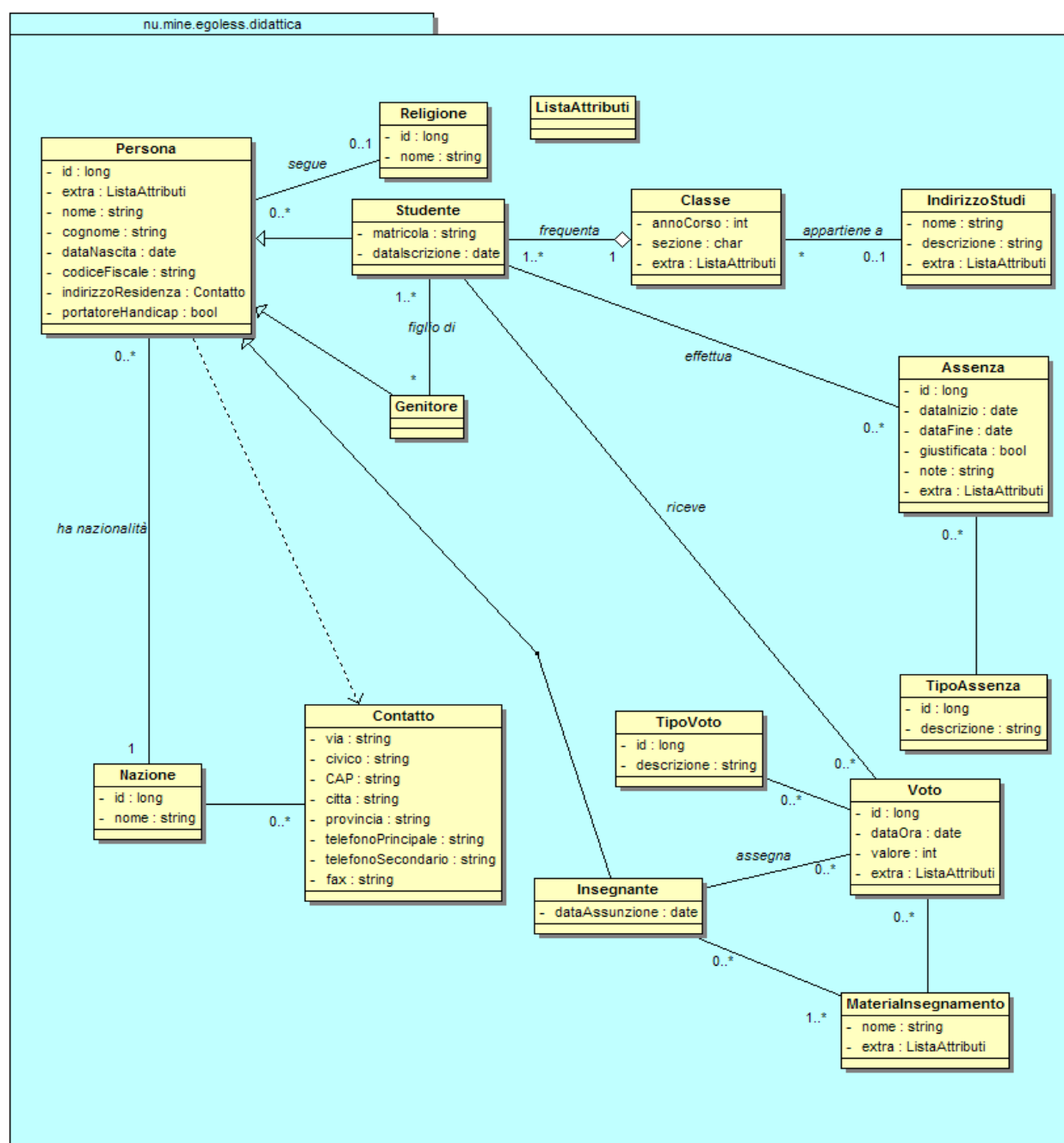


Illustrazione 5: Diagramma delle classi del WS-Didattica

- **classi per la ricerca.** Sono strutture dati che contengono tutti i parametri per eseguire una data ricerca su oggetti di classi complesse. Sono previste per poter

modificare agevolmente i criteri di una ricerca. Raccogliere tutti i criteri di una ricerca in una classe, piuttosto che lasciarli come argomenti della funzione di ricerca, permette di aggiungere o cambiare i criteri di ricerca senza alterare i parametri della funzione (agevolando così la manutenzione).

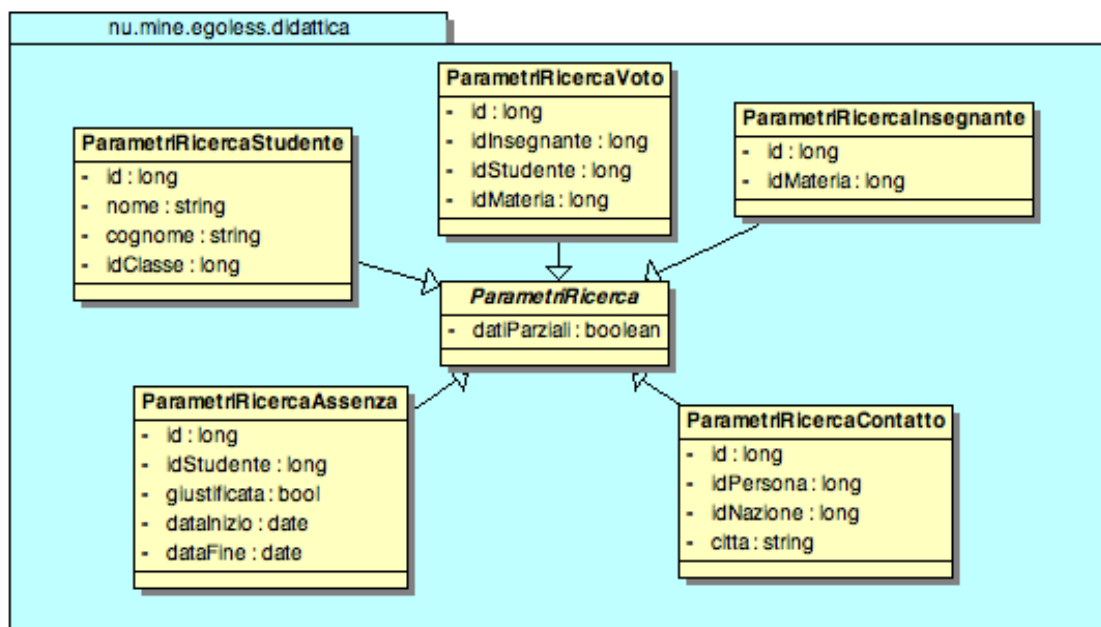


Illustrazione 6: Diagramma delle classi per la ricerca

Ogni classe maneggiata presenta un attributo id che identifica univocamente tutti i suoi oggetti. Questo permette di avere un'equivalente del puntatore ad un oggetto a livello di Web Service e di rendere efficiente l'invio di messaggi (per esempio, per la cancellazione di un elemento basterà indicare il suo id e non ripassare inutilmente tutto l'elemento).

Per ogni classe complessa esposta vengono forniti i seguenti metodi nell'interfaccia del Web Service:

- aggiunta di un elemento;
- modifica di un elemento. Abbiamo preferito separare le operazioni di aggiunta e modifica per rendere più chiara l'interfaccia del Web Service;
- rimozione di un elemento esistente;
- ricerca di un insieme di elementi in base a criteri. In questo modo si cerca di minimizzare il traffico tra applicazione e il Web Service.

Per ogni classe semplice vengono invece forniti i seguenti metodi:

- aggiunta di un nuovo elemento;
- rimozione di un elemento esistente;
- recupero di tutti gli elementi.

Le ragioni di questa scelta sono:

- Dato che le classi semplici contengono pochi (due/tre attributi) e stabili nel tempo (es. Tipi di giustificazioni, Tipi Assenze) non è necessario avere un metodo di modifica. Se un oggetto contiene dati da modificare è più conveniente cancellarlo e crearne uno nuovo con i dati corretti.
- Non serve un metodo di ricerca per le classi semplici. Considerato che gli oggetti di una classe semplice sono pochi, ogni oggetto contiene pochi attributi e stabili nel tempo, è sensato caricare tali dati all'avvio dell'applicazione.

Per le classi semplici (Nazione o Religione) per cui le istanze sono già decise è presente solo l'ultimo metodo che recupera tutti gli oggetti e non da la possibilità di modificarli per cui le istanze sono già decise è presente solo l'ultimo metodo.

La documentazione dei Web Service è presente in [DocIWS].

La documentazione delle classi di ricerca è presente in [DocCRIC].

Le rimanenti classi sono documentate qui di seguito. Per ogni classe si ometterà l'attributo id.

CSW-01/1 - Extra - classe	
<i>Breve descrizione:</i>	Questa classe rappresenta un insieme di informazioni aggiuntive generiche. Probabilmente sarà un insieme di oggetti o di stringhe. Viene impiegato nelle classi complesse che potrebbero variare di struttura nel tempo a causa del cambiamento di requisiti. Extra non è ne' classe semplice ne' classe complessa.
<i>Attributi:</i>	-
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/3 - Nazione - classe semplice	
<i>Breve descrizione:</i>	Rappresenta una nazione. Questa classe è importante per fissare i valori validi per i campi Nazione.

<i>Attributi:</i>	<ul style="list-style-type: none"> • nome: nome della nazione
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Persona tramite “ha nazionalità”: una persona è di una certa nazionalità • Contatto: la classe Contatto viene usata per memorizzare i contatti

CSW-01/4 - Religione - classe semplice

<i>Breve descrizione:</i>	Rappresenta una religione. Questa classe è importante per fissare i valori validi per i campi Religione.
<i>Attributi:</i>	<ul style="list-style-type: none"> • nome: nome della religione.
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Persona tramite “segue”: una persona crede o meno in una religione

CSW-01/5 - Contatto - classe semplice

<i>Breve descrizione:</i>	Racchiude le informazioni su come raggiungere una persona (principalmente indirizzo e telefono).
<i>Attributi:</i>	<ul style="list-style-type: none"> • via: via • civico: il numero dell'abitazione • CAP: Codice di Avviamento Postale • città: città • provincia: provincia • telefonoPrincipale: numero di telefono primario • telefonoSecondario: altro numero di telefono • fax: numero di fax
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Persona: la classe descritta viene usata per memorizzare i contatti di una persona

CSW-01/6 - Studente - classe complessa

<i>Breve descrizione:</i>	Rappresenta uno studente della scuola.
<i>Attributi:</i>	<ul style="list-style-type: none"> • Matricola: numero identificativo univoco per lo studente • dataIscrizione: data di iscrizione alla scuola
<i>Eredita da:</i>	<ul style="list-style-type: none"> • Persona
<i>Legata a:</i>	<ul style="list-style-type: none"> • Classe tramite “frequenta”: uno studente

	<p>frequenta una classe</p> <ul style="list-style-type: none"> Assenza tramite “effettua”: uno studente può fare un certo numero di assenze (anche zero) Voto tramite “riceve”: un utente riceve una votazione da un professore
--	---

CSW-01/7 - Classe - classe semplice

<i>Breve descrizione:</i>	Rappresenta una classe come insieme di alunni.
<i>Attributi:</i>	<ul style="list-style-type: none"> annoCorso: la numerazione della classe si identifica con l'anno del corso sezione: sezione della classe extra: informazioni extra
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> IndirizzoStudi tramite “appartiene a”: una classe appartiene ad un indirizzo di studi Studente tramite “frequenta”: una classe è composta da almeno uno studente

CSW-01/8 - Voto - classe complessa

<i>Breve descrizione:</i>	Rappresenta una valutazione assegnata ad uno studente in una materia.
<i>Attributi:</i>	<ul style="list-style-type: none"> id: identificativo univoco dataOra: data in cui è stato registrato il voto valore: valore numerico del voto extra: informazioni aggiuntive
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> Studente tramite “riceve”: un voto è assegnato ad uno studente TipoVoto: un voto può essere di un certo tipo Insegnante tramite “assegna”: un insegnante assegna un voto MaterialInsegnamento: un voto è assegnato per una data materia

CSW-01/9 - TipoVoto - classe semplice

<i>Breve descrizione:</i>	Specifica il tipo di valutazione. E' pensato per distinguere i normali voti dai voti di fine periodo (trimestre, quadrimestre) e fine anno.
<i>Attributi:</i>	<ul style="list-style-type: none"> id: identificativo univoco descrizione: descrizione del tipo di valutazione
<i>Eredita da:</i>	-

<i>Legata a:</i>	<ul style="list-style-type: none"> Voto: ogni voto ha un certo tipo di valutazione
------------------	---

CSW-01/10 - MaterialInsegnamento - classe semplice

<i>Breve descrizione:</i>	Rappresenta una materia.
<i>Attributi:</i>	<ul style="list-style-type: none"> nome: nome della materia extra: informazioni aggiuntive
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> Insegnante: un insegnante è abilitato in almeno una materia Voto: un voto è assegnato per una data materia

CSW-01/11 - Insegnante - classe complessa

<i>Breve descrizione:</i>	Rappresenta un professore della scuola.
<i>Attributi:</i>	<ul style="list-style-type: none"> dataAssunzione: data di assunzione
<i>Eredita da:</i>	<ul style="list-style-type: none"> Person
<i>Legata a:</i>	<ul style="list-style-type: none"> Voto tramite “assegna”: un insegnante può dare più voti MaterialInsegnamento: un insegnante è abilitato in almeno una materia

CSW-01/12 - Assenza - classe complessa

<i>Breve descrizione:</i>	Rappresenta l'assenza di uno studente.
<i>Attributi:</i>	<ul style="list-style-type: none"> id: identificativo univoco dataInizio: giorno/ora in cui è iniziata l'assenza dataFine: giorno/ora in cui è finita l'assenza giustificata: indica se l'assenza è stata giustificata o meno note: annotazioni aggiuntive extra: informazioni aggiuntive
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> TipoAssenza: una assenza è di un dato tipo Studente tramite “effettua”: uno studente può fare un certo numero di assenze (anche zero)

CSW-01/13 - TipoAssenza - classe semplice

<i>Breve descrizione:</i>	Rappresenta una tipologia di assenza. Si è ritenuto opportuno introdurre questa classe per catalogare e
---------------------------	---

Egoless Group

	rendere immediatamente fruibili le varie tipologie di assenza (malattia, motivi sportivi, ...).
<i>Attributi:</i>	<ul style="list-style-type: none">• id: identificativo univoco• descrizione: descrizione della tipologia di assenza
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none">• Assenza: una assenza è di un dato tipo

4 Stime di fattibilità e di bisogno di risorse

Per l'impegno ed il tempo richiesto alla realizzazione, fare riferimento a [PP].

Per la realizzazione delle componenti, si sono individuate le seguenti tecnologie:

- Java 6;
- Java EE 5, in particolare JAX-WS 2.0;
- Glassfish 1.0 (equivalente a Sun Java System Application Server Platform Edition 9).

5 Tracciamento della relazione componenti – requisiti

Matrice? Tracciamento descrittivo come SGPEM?

\	CSA-01 / CSW-01 Didattica	CSA-02 / CSW-02 Documenti	CSA-03 / CSW-03 Monitoraggio	CSA-04 / CSW-04 Accesso telematico
01	x			
02	x			
03	x			
04		x		
05		x		
06			x	
07				
08				x
09		x		
10		x		
11			x	
12	x			
13	x			
14	x			
15				
16		x		
17	x			

\	CSA-01 / CSW-01 Didattica	CSA-02 / CSW-02 Documenti	CSA-03 / CSW-03 Monitoraggio	CSA-04 / CSW-04 Accesso telematico
18				
19				x
20			x	
21	x			
22	x			
23				x
24		x		
25	x			
26			x	
27			x	
28				
29				x
30			x	
31			x	
32	x			