

Indice

1	I requisiti di un'applicazione per servizi universitari	2
1.1	I casi d'uso	6
1.2	Conclusioni	30
2	Progettazione di un'applicazione per servizi universitari	31
2.1	Introduzione	31
2.2	Analisi dei casi d'uso	31
2.3	Progettazione	36
2.3.1	Piano di studi	37
2.3.2	Tirocini	43
2.3.3	Prova finale	54
A	Appendice	64

Capitolo 1

I requisiti di un'applicazione per servizi universitari

In questo capitolo vengono analizzati i requisiti per la costruzione di un'applicazione Web per un corso di studi. Sono stati quindi individuati alcuni servizi che si ritengono utili a migliorare l'organizzazione di un sito Internet per corso di studi. Tra questi, in particolare, si studierà la gestione di alcuni processi: tirocini, tesi, piani di studi.

Per far ciò si utilizzano i seguenti strumenti:

- suddivisione degli utenti in classi, in conformità a Sheidelmann [S 92]
- dizionario di progetto, in conformità a Conallen [JC 02]
- elenco dei requisiti funzionali, in conformità a Conallen [JC 02]
- descrizione per scenario di sequenza, in conforminta a Conallen [JC 02] e Fowler [FS 00]
- rappresentazione dei diagrammi dei casi d'uso, in conformità a Conallen [JC 02]
- diagrammi di sequenza ad alto livello, in conformità a Conallen [JC 02].

Per prima cosa, come si è visto, è importante suddividere le classi di utenti a seconda delle funzioni che andranno a svolgere con l'applicazione. Le classi individuate sono:

1. **gli studenti:** interessati a tutte le informazioni che riguardano il corso di studi, in particolar modo gli orari, i programmi, e il materiale didattico di ogni materia, possono anche essere interessati agli orari dei laboratori, delle biblioteche, di segreteria ecc. Inoltre potrebbero avere la possibilità di decidere via Web il piano di studi, il tirocinio da svolgere e il titolo della prova finale. Navigando per la rete si scopre che alcuni studenti di Scienze di Internet dell'Università di Bologna si sono costruiti due siti di community (www.menesbatta.com/www.i-sciences.net), in cui si scambiano notizie e pareri sul corso di studi e su altre attività di loro interesse. Da questi siti si può capire che comunque gli studenti hanno interesse a comunicarsi informazioni.
2. **le famiglie:** negli ultimi due anni è aumentato il numero di genitori che si occupano in prima persona dell'iscrizione dei figli a un corso universitario. Quindi si potrebbe cercare di creare una sezione per l'orientamento, dove vengono presentati il corso di studi, gli sbocchi professionali, la percentuale di laureati che trovano lavoro ecc.
3. **il personale docente:** potrebbe servirsi del sito Web per presentare il proprio curriculum vitae, le attività di ricerca in cui è coinvolto, il corso o i corsi di cui è docente. Potrebbe essere utile anche un sistema che gestisca le iscrizioni agli esami via Web e avere a disposizione uno spazio dove inserire i titoli o gli argomenti delle tesi disponibili per gli studenti.
4. **le aziende:** possono essere interessate a convenzionarsi con l'università per ospitare studenti a scopo tirocinio. Una volta convenzionate, potrebbero proporre i loro progetti formativi attraverso un servizio Internet.
5. **il personale non docente:** potrebbe servirsi del sito per dare informazioni sugli orari di segreteria, istruzioni agli studenti su alcune pratiche burocratiche e rendere disponibili moduli.

Il presente lavoro dedica maggiore attenzione alle parti di progettazione e analisi che riguardano alcuni servizi di gestione del corso di studi. Questi servizi sono di sicura utilità e possono diminuire i cosiddetti tempi burocratici. È interessante comunque - per un ulteriore miglioramento del servizio di un sito Web universitario - mantenere anche l'ipotesi di introdurre una parte dedicata ai servizi di comunità, di scambio tra studenti, di iscrizione agli esami, di pagine per i docenti.

Come si è visto nel capitolo precedente, uno dei punti fondamentali per portare avanti un progetto è avere ben chiari gli obiettivi, soprattutto quando si lavora in gruppo. L'obiettivo del progetto è rendere disponibile via Web servizi che riguardano la richiesta di convenzione delle aziende, l'accettazione della convenzione da parte del corso di studi, l'invio di proposte di progetti formativi e la loro approvazione, la raccolta delle richieste di tirocinio, la presentazione dei piani di studio e la loro approvazione da parte del Consiglio di corso di studi, infine la consultazione di un elenco di titoli di tesi e l'archiviazione delle tesi completate.

Analizzando gli obiettivi più schematicamente, si possono quindi individuare i requisiti funzionali del sistema:

1. gestione dei piani di studi
2. gestione dei tirocini
3. gestione delle prove finali
4. gestione delle convenzioni
5. gestione dei progetti formativi.

Prima di iniziare l'analisi vera e propria conviene stilare un elenco dei termini (figura 4.1 e 4.2) che si prevede di utilizzare con più frequenza nel corso del progetto. L'elenco e le relative definizioni giustificano la loro banalità con la necessità di evitare equivoci di ogni sorta, considerando che nel progetto potrebbero lavorare molte persone diverse.

<i>Termine</i>	<i>Definizione</i>	<i>Sinonimo</i>
Prova finale	La prova conclusiva che uno studente deve sostenere per laurearsi	Tesi
Piano di studi	Modulo in cui gli studenti indicano gli esami opzionali che vogliono sostenere	
Carriera	Indica la storia di uno studente, quindi gli esami sostenuti, gli esami che deve sostenere, i voti che ha ottenuto ecc.	
Titolo prova finale	Ogni prova finale è un elaborato che lo studente deve preparare, i titoli vengono proposti da un docente (relatore)	Titolo della Tesi
Relatore	Il relatore è il docente che segue lo studente durante la redazione della prova finale	
Settore disciplinare	E' la classe di appartenenza di una materia	
Esame	Prova conclusiva di un insegnamento	
Crediti formativi	Un numero che indica lo sforzo in quantità di tempo che uno studente deve fare per poter superare l'esame	Crediti
Materia	Nome con cui viene indicato un insegnamento	Insegnamento
Convenzione	Accordo tra un'azienda e l'università (solo dopo che un'azienda si è convenzionata può ospitare dei tirocinanti).	
Progetto formativo	L'attività che lo studente deve svolgere durante il periodo di tirocinio	
Tirocinio	Periodo di circa tre mesi svolto presso un'azienda	
Studente	Colui che iscritto all'università	Iscritto, Immatricolato
Docente	Colui che insegna all'università	Professore

Figura 1.1: glossario dei termini

<i>Termine</i>	<i>Definizione</i>	<i>Sinonimo</i>
Tesista	Studente che può svolgere o che sta svolgendo la tesi	
Impiegato	Colui che lavora per l'università e non svolge attività didattica	
Tirocinante	Studente che svolge il tirocinio o che può farne richiesta	
Metadati	Dati che servono per alcune informazioni sulla prova finale	
Consiglio di corso studi (Ccds)	E' il consiglio dei professori di un corso di studi: questo organo ha potere decisionale per questioni quali i piani di studio, i tirocini, convenzioni, ecc.	Consiglio di corso di laurea (Ccdl)

Figura 1.2: glossario dei termini

1.1 I casi d'uso

1. Consegna del piano di studi

Dichiarazione degli obiettivi: lo studente può compilare il modulo del piano di studio via Web, e può inviarlo direttamente al Consiglio di Corso di Studi, che deciderà se approvarlo o meno.

Descrizione:

- a. uno studente si collega al sito
- b. lo studente si identifica
- c. il sistema controlla che lo studente esista
- d. il sistema verifica che lo studente possa compilare il piano di studi
- e. lo studente compila il modulo di piano di studi

- f. il sistema controlla che il piano di studio sia completo nelle sue parti obbligatorie
- g. il sistema inoltra a un elenco di richieste di piani di studi.

Alternativa: *lo studente non è riconosciuto*

Al passo c. il sistema non riconosce lo studente.

Si deve permettere di riprovare a identificarsi.

Alternativa: *lo studente non può consegnare il piano di studi*

Al passo d. se il sistema verifica che lo studente non può presentare il piano di studi (perché non ha i crediti, perché non è nei tempi giusti, ecc.) a seconda del regolamento del corso di studi)

Si restituisce un messaggio di errore.

Alternativa: *lo studente non ha compilato tutte le parti del piano di studio*

Al passo f. se il sistema si accorge che il modulo non è stato compilato in almeno un campo obbligatorio.

Si deve permettere di inserire le informazioni mancanti.

Priorità: media

Ipotesi: il sistema deve avere un elenco di piani di studi da approvare

Precondizioni: lo studente può fare richiesta del piano di studi

Postcondizioni: il piano di studi è completo in tutte le sue parti obbligatorie e rispetta le regole date dal regolamento didattico.

Problemi aperti:

Requisiti soddisfatti: 1

Diagramma dei casi d'uso: figura 4.3

Diagramma di sequenza: figura 4.4

2. Approvazione del piano di studi

Dichiarazione degli obiettivi: il Consiglio di corso di studi controlla la carriera di ogni studente che ha presentato il piano di studi. Se approva il piano non viene fatta nessuna comunicazione, altrimenti viene avvertito lo studente che deve modificare alcune parti del piano.

Descrizione:

- a. il Consiglio di corso di studi accede all'elenco dei piani di studi da approvare
- b. il sistema affianca a ogni elemento dell'elenco e i dati dello studente utili per approvare un piano di studi
- c. il Consiglio di corso di studi decide se approvare o meno il piano di studi
- d. il Consiglio di corso di studi inserisce il piano di studi approvato

Alternativa: il Consiglio non approva il piano di studi

Al passo c. il Consiglio di corso di studi non approva il piano di studi

Viene avvertito lo studente che occorre rifare il piano di studi

Priorità: alta

Ipotesi: il sistema possiede un elenco di richieste

Precondizioni: solo il Consiglio di corso di studi può accedere all'elenco dei piani di studi

Postcondizioni: lo studente sa se gli è stato riconosciuto o meno il piano di studi

Problemi aperti:

Requisiti soddisfatti: 1

Diagramma dei casi d'uso: figura 4.3

Diagramma di sequenza: figura 4.4

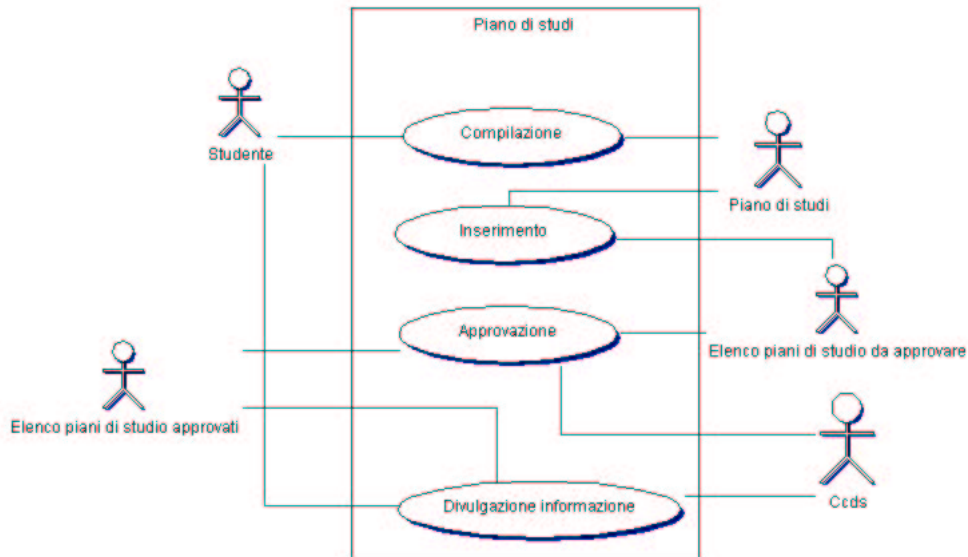


Figura 1.3: piano di studi

Diagrammi di sequenza: di seguito sono riportati i due diagrammi di sequenza. Il primo si riferisce (fig. 4.4) al caso d'uso 1 (Consegna del piano di studio) e 2 (Approvazione dei piani di studi). Dopo che è stato riconosciuto, lo studente compila il piano di studi, il sistema controlla che siano stati inseriti tutti i campi fondamentali. Nel caso in cui sia stato compilato correttamente, il piano viene inserito nell'elenco dei piani di studi da approvare, altrimenti viene avvertito lo studente per apportare modifiche. Il Consiglio di corso di studi, in una data prefissata dopo il termine finale della consegna dei piani di studi, controlla i piani e decide se approvarli o meno.

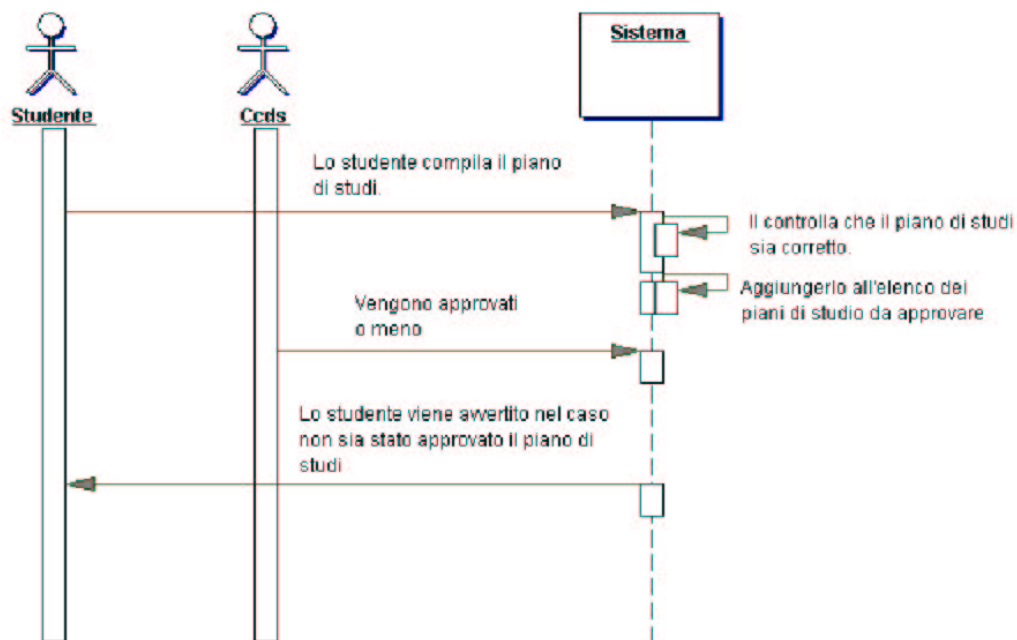


Figura 1.4: piano di studi

3. Richiesta tirocinio

Dichiarazione degli obiettivi: lo studente, dopo essersi informato sui progetti formativi nel sito Web del suo corso di studi, può scegliere quale tirocinio preferisce; la richiesta viene poi inserita nell'elenco dei tirocini da approvare.

Descrizione:

- a. lo studente consulta l'elenco dei progetti formativi
- b. lo studente si identifica al sistema
- c. il sistema controlla che lo studente esista
- d. il sistema controlla che lo studente possa fare il tirocinio
- e. lo studente compila la domanda di richiesta
- f. il sistema controlla che il modulo sia compilato correttamente

- g. il sistema controlla che, se nel progetto formativo scelto dallo studente vengono richiesti determinati esami, questi siano stati realmente sostenuti dallo studente
- h. il sistema inserisce la richiesta in un elenco di richieste
- i. il sistema modifica le statistiche di richiesta di quel determinato progetto formativo

Alternativa: *lo studente non viene identificato*

Passo c. se il sistema non riconosce lo studente permette di tentare di identificarsi nuovamente.

Si deve permettere di riprovare a identificarsi.

Alternativa: *lo studente non può fare richiesta di tirocinio*

Al passo d. il sistema riscontra la non idoneità dello studente a fare richiesta di tirocinio.

Il sistema nega allo studente la possibilità di entrare nella pagina per compilare la richiesta di tirocinio.

Alternativa: *lo studente non ha compilato correttamente il modulo*

Al passo f. il sistema non accetta la domanda perché non è stata compilata correttamente.

Si deve permettere di correggere la domanda

Alternativa: *lo studente non ha fatto tutti gli esami richiesti*

Al passo g. il sistema si accorge che lo studente non può svolgere quel progetto formativo perché non ha sostenuto tutti gli esami necessari.

Il sistema chiede allo studente di modificare la sua scelta.

Priorità: media

Ipotesi: il sistema possiede un elenco di progetti formativi

Precondizioni: lo studente può svolgere l'attività di tirocinio

Postcondizioni: lo studente ha fatto richiesta per il tirocinio ed è in attesa di risposta

Problemi aperti: Controllare i tempi di consegna delle richieste

Requisiti soddisfatti: 2

Diagramma dei casi d'uso: figura 4.5

Diagramma di sequenza: figura 4.6

4. Assegnazione tirocinio

Dichiarazione degli obiettivi: il Consiglio di corso di studi assegna a ogni studente che ne ha fatto richiesta un tirocinio.

Descrizione:

- a. il Consiglio di corso di studi assegna i progetti formativi
- b. il sistema elimina i progetti formativi assegnati dall'elenco dei progetti formativi
- c. il sistema azzera le statistiche dei progetti formativi non assegnati

Priorità: alta

Ipotesi: il sistema possiede un elenco delle richieste

Precondizioni: almeno uno studente ha fatto domanda di tirocinio

Postcondizioni: i tirocini sono stati assegnati e gli studenti possono iniziare il tirocinio

Problemi aperti:

Requisiti soddisfatti: 2

Diagramma dei casi d'uso: figura 4.5

Diagramma di sequenza: figura 4.6

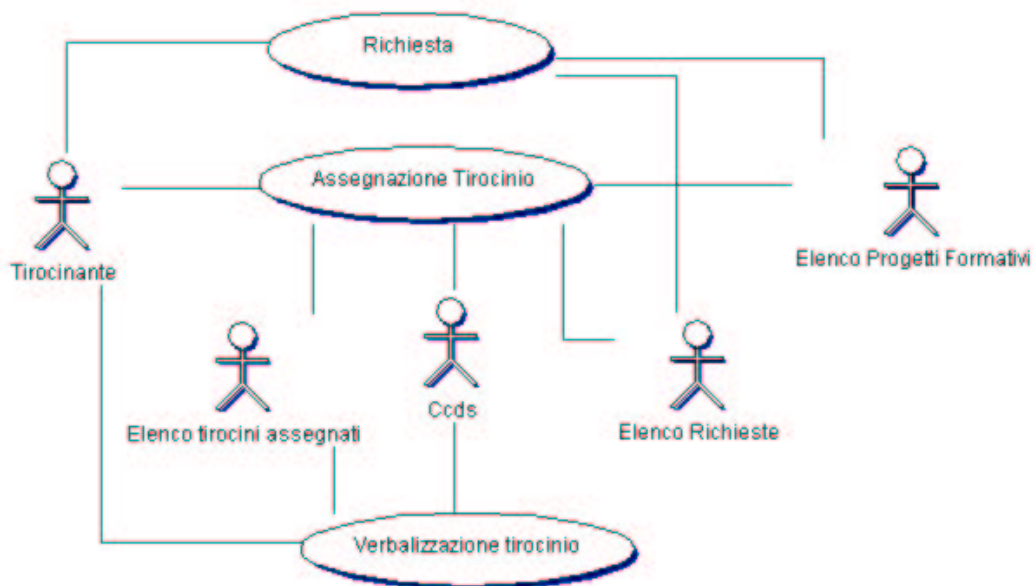


Figura 1.5: richiesta e assegnazione tirocinio

Il diagramma di sequenza: per prima cosa lo studente si informa sui progetti formativi che può scegliere; dopo che è stato identificato, lo studente può richiedere il tirocinio. Se il sistema glielo permette può immettere la sua richiesta. Si controlla che gli esami sostenuti dallo studente siano conformi agli esami richiesti per svolgere il progetto formativo scelto. Se viene accettata, la domanda viene inserita nell'elenco richieste tirocinio. Nei termini definiti dal corso di studi il Consiglio di corso di studi assegna i tirocini. I tirocini assegnati vengono inseriti nell'elenco tirocini attivi, viene poi eliminato il progetto formativo dall'elenco progetti formativi. Qui di seguito viene mostrato il diagramma di sequenza di richiesta e assegnazione dei tirocini.

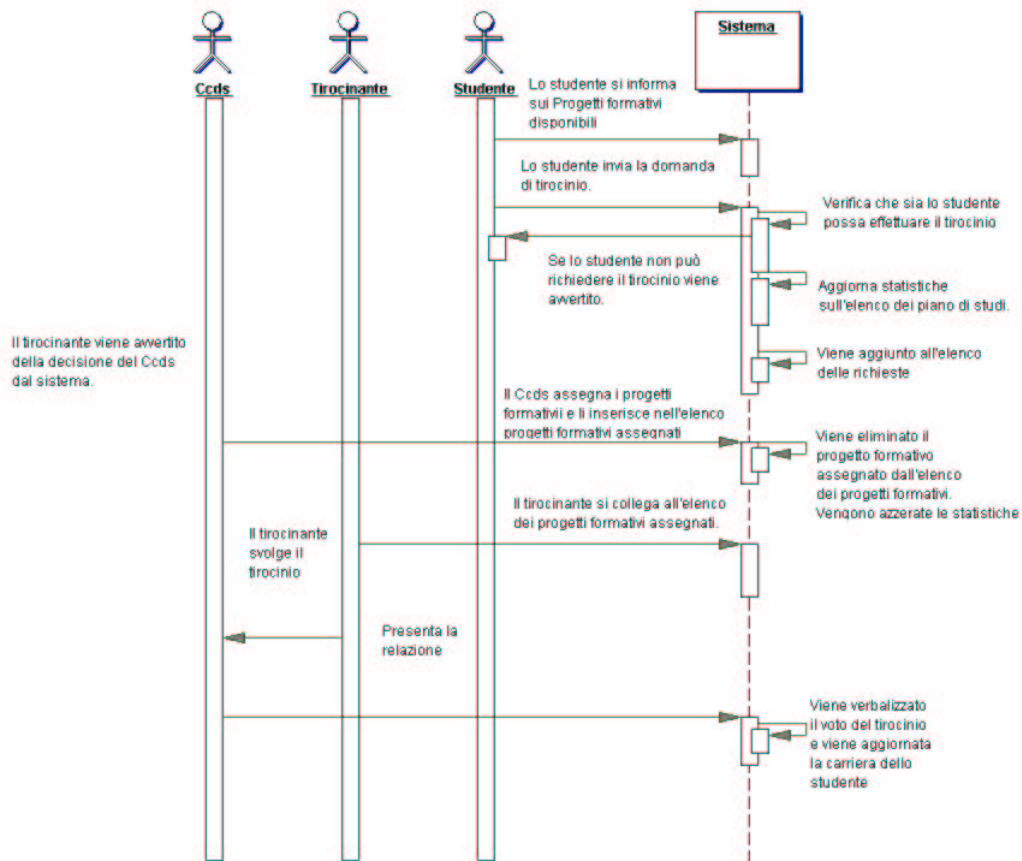


Figura 1.6: richiesta e assegnazione tirocinio

5. scelta del titolo della prova finale

Dichiarazione degli obiettivi: lo studente che è al termine dei suoi studi sceglie una prova finale tra quelle proposte dai professori.

Descrizione:

- lo studente si informa nel sito su quali proposte sono disponibili
- lo studente va a ricevimento dal docente proponente la prova finale
- lo studente decide se affrontare o meno la prova finale.

Alternativa: *lo studente non sceglie la prova finale propositagli*

Al passo c. allo studente non piace la proposta

Lo studente cerca una nuova proposta

Priorità: bassa

Ipotesi: il sistema possiede un elenco di titoli di prove finali

Precondizioni: lo studente si deve autenticare

Postcondizioni: lo studente ha scelto un titolo

Problemi aperti:

Requisiti soddisfatti: 3

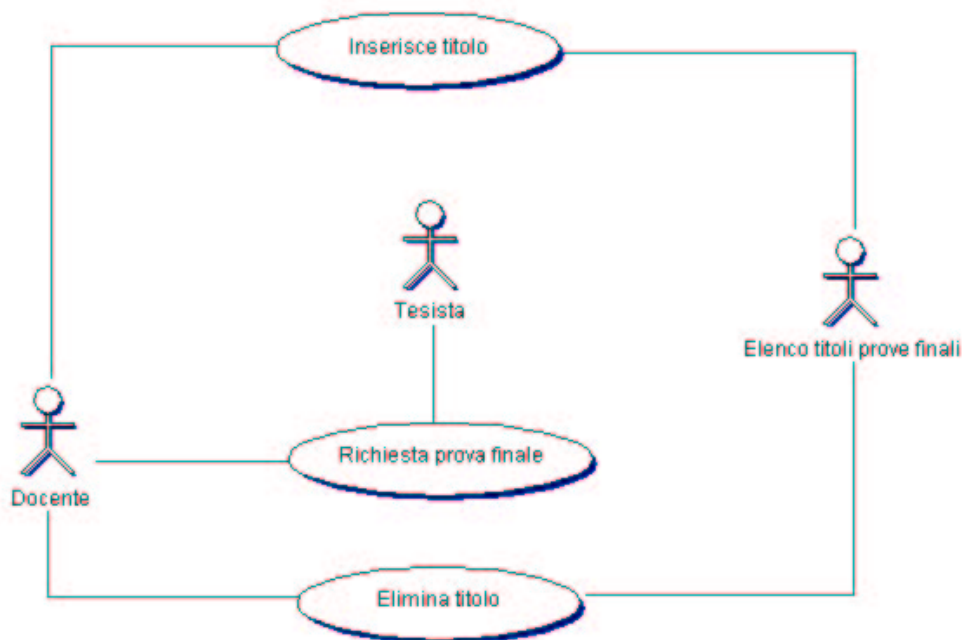


Figura 1.7: scelta delle prove finali

Il diagramma di sequenza: non è qui illustrato perché viene incluso nel caso d'uso dell'eliminazione titoli della prova finale da parte dei docenti (figura 4.16).

6. Consegna della prova finale e inserimento in archivio della stessa

Introduzione

Per la progettazione di questo caso d'uso è utile tener presente che esiste un sito che prevede l'archiviazione delle tesi. Il sito è www.ndltd.org [W2].

Nel sito Internet della NDLTD (Networked Digital Library of Theses and Dissertation) vengono spiegati alcuni standard di metadati per l'archiviazione delle tesi. I metadati sono delle informazioni generiche sulla prova finale come ad esempio il titolo, l'oggetto, la materia ecc. Si affronterà con maggiore attenzione questo argomento nella fase di progettazione.

Dichiarazione degli obiettivi: lo studente consegna la prova finale sia in formato cartaceo sia in formato elettronico. Nel formato elettronico deve essere presente anche un file che racchiuda delle informazioni sulla tesi. L'impiegato della segreteria provvederà, una volta che la prova finale è stata discussa, a inserirla nell'archivio.

Descrizione:

- a. lo studente consegna la tesi
- b. lo studente discute la tesi
- c. l'impiegato inserisce i metadati e la tesi
- d. il sistema controlla la coerenza tra contenuto della tesi e i metadati

Alternativa: *i metadati non rispecchiano il contenuto della tesi*

Al passo d., quando il sistema controlla la tesi e li confronta con i metadati, essi risultano non coerenti.

Il sistema non inserisce la tesi e l'impiegato avverte lo studente dell'incoerenza.

Priorità: media

Ipotesi: il sistema ha un archivio di prove finali e di metadati

Precondizioni: la tesi deve essere consegnata in formato elettronico

Postcondizioni: la tesi è pronta per essere consultata

Problemi aperti:

Requisiti soddisfatti: 3

Diagramma dei casi d'uso: figura 4.8

Diagramma di sequenza: figura 4.9

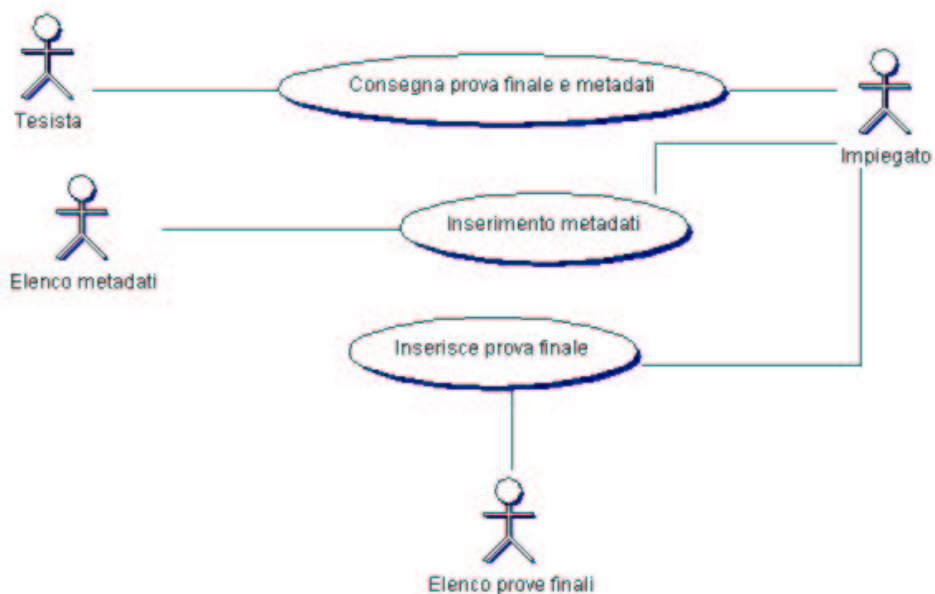


Figura 1.8: inserimento della prova finale in archivio

Il diagramma di sequenza: il tesista consegna la tesi all'impiegato che successivamente alla discussione dello studente inserisce metadati e tesi nell'archivio del sistema; a questo punto gli utenti esterni possono consultare l'archivio.

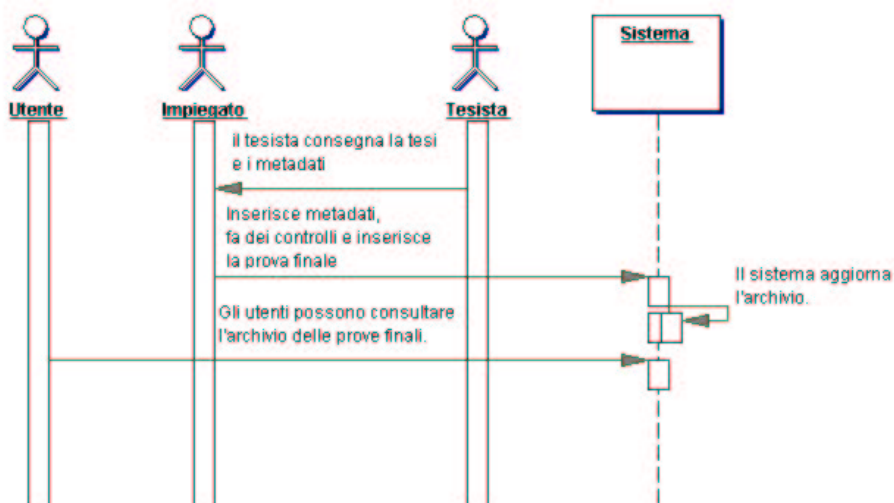


Figura 1.9: inserimento della prova finale in archivio

7. Richiesta di convenzione

Dichiarazione degli obiettivi: le aziende fanno richiesta di convenzione attraverso un'applicazione Web, o ne richiedono la disdetta.

Descrizione:

- a. l'azienda si informa sul corso di studi
- b. l'azienda richiede di essere convenzionata
- c. il sistema controlla che siano stati inseriti tutti i campi obbligatori
- d. il sistema inserisce la domanda nella lista delle richieste di convenzione

Alternativa: *l'azienda non ha inserito tutti i campi obbligatori*

Al passo c. il sistema verifica che almeno uno dei campi obbligatori non è stato compilato.

Il sistema richiede i dati mancanti.

Priorità: alta

Ipotesi: il sistema ha un elenco delle aziende che richiedono la convenzione

Precondizioni: l'azienda è interessata alla convenzione

Postcondizioni: l'azienda è in attesa di una risposta da parte del Consiglio di corso di studi

Problemi aperti:

Requisiti soddisfatti: 4

Diagramma dei casi d'uso: figura 4.10

Diagramma di sequenza: figura 4.11

8. Convenzione di un'azienda

Dichiarazione degli obiettivi: Il Consiglio di corso di studi sceglie se convenzionarsi o meno con un'azienda. La scelta viene fatta grazie ai dati che l'azienda ha fornito per richiedere la convenzione.

Descrizione:

- a. il Corso di studi accede all'elenco delle richieste di convenzione
- b. il Corso di studi decide se convenzionare l'azienda

- c. l'azienda viene inserita in elenco o viene avvertita del rifiuto
- d. il corso di studi attende i documenti
- e. l'azienda viene inserita nell'elenco aziende convenzionate e riceve un codice di convenzione

Alternativa: *viene respinta la richiesta di convenzione*

Al passo b. il Consiglio di corso di studi decide di non convenzionare l'azienda.

L'azienda viene avvertita del rifiuto.

Alternativa: *non si ricevono i documenti dall'azienda*

Al passo d. l'azienda non invia i documenti.

Viene contattata l'azienda per accertamenti: nel caso non voglia più convenzionarsi, viene eliminata dall'elenco aziende in attesa di convenzione.

Priorità: alta

Ipotesi: il sistema ha un elenco dell'aziende che richiedono la convenzione

Precondizioni: il corso di studi ha bisogno di avere delle aziende convenzionate

Postcondizioni: l'azienda è convenzionata

Problemi aperti:

Requisiti soddisfatti: 4

Diagramma dei casi d'uso: figura 4.10

Diagramma di sequenza: figura 4.11

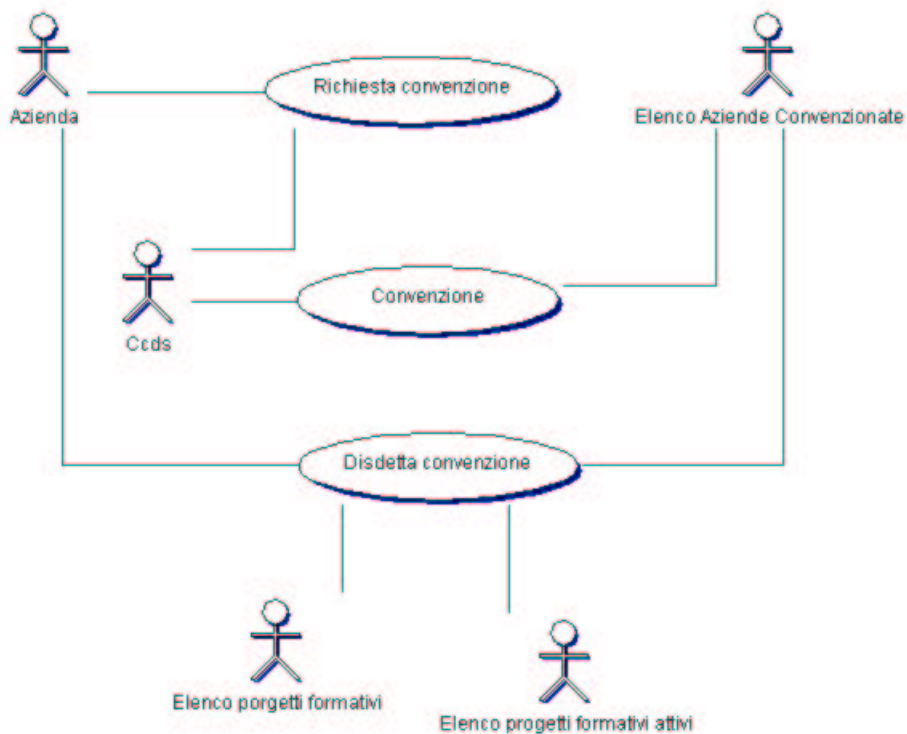


Figura 1.10: richiesta accettazione ed eliminazione della convenzione

Il diagramma di sequenza: l'azienda invia la richiesta di convenzione al sistema, che invia la richiesta al Consiglio di corso di studi; il Consiglio di corso di studi dopo aver preso una decisione avverte l'azienda; se è stata accettata la richiesta, il sistema immette i dati dell'azienda nell'elenco aziende in attesa di convenzione fino a quando non riceve i documenti necessari per concludere la pratica; il Consiglio di corso di studi ricevuti i documenti, inserisce l'azienda nell'elenco delle aziende convenzionate e invia all'azienda il codice di convenzione, che servirà per inserire le proposte di progetto formativo.

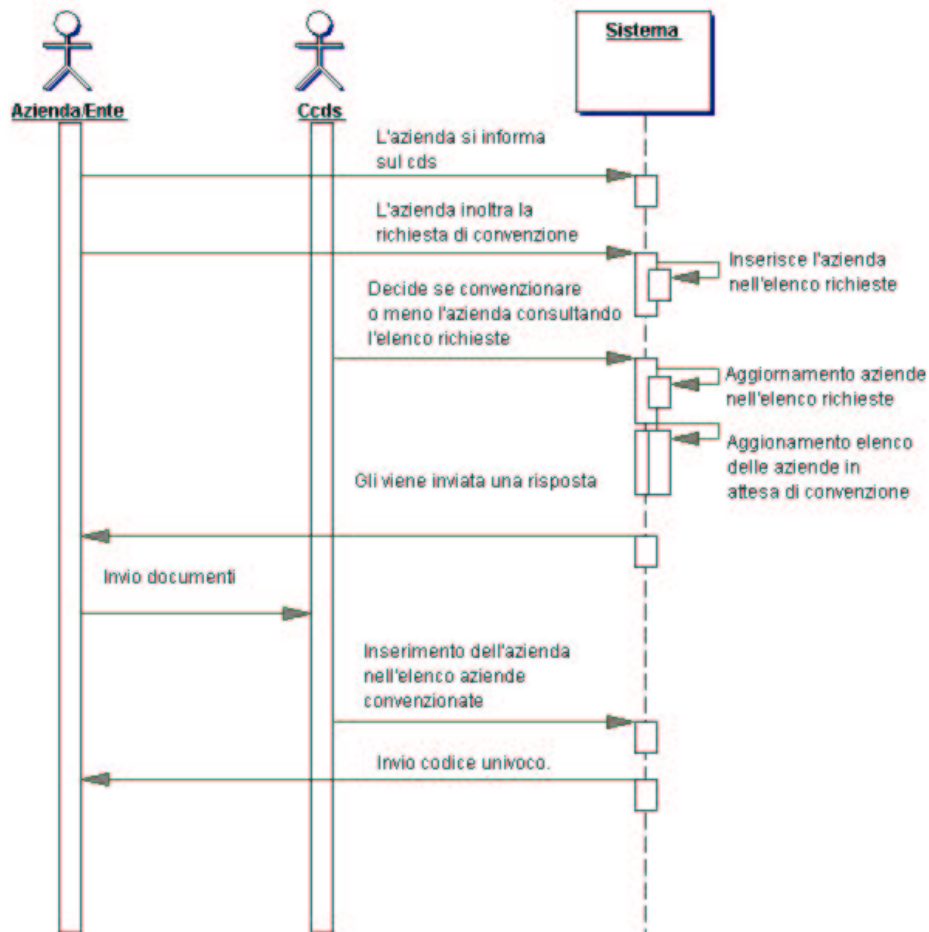


Figura 1.11: richiesta e accettazione convenzione

9. Disdetta della convenzione

Dichiarazione degli obiettivi: l'azienda desidera disdire la convenzione

Descrizione:

- a. l'azienda si identifica
- b. l'azienda richiede di eliminare il suo nominativo dall'elenco delle aziende convenzionate
- c. il sistema controlla i progetti formativi attivi
- d. il sistema controlla le statistiche dell'elenco delle richieste di tirocinio.

- e. vengono eliminati tutti i progetti formativi dell'azienda
- f. viene eliminata l'azienda dall'elenco delle convenzioni

Alternativa: *l'azienda non viene riconosciuta*

Al passo a. l'azienda non viene riconosciuta dal sistema.

Il sistema le richiede di identificarsi.

Alternativa: *l'azienda ha dei tirocini attivi*

Al passo c. l'azienda risulta avere dei tirocini attivi.

Viene rifiutato l'annullamento della convenzione.

Alternativa: *l'azienda ha delle proposte di tirocinio con statistiche diverse da zero*

Al passo d. l'azienda risulta avere dei progetti formativi con statistica diversa da zero.

Viene rifiutato l'annullamento della convenzione.

Priorità: alta

Ipotesi: il sistema ha un elenco dell'aziende convenzionate, dei tirocini attivi e dei progetti formativi

Precondizioni: l'azienda non vuole più ospitare dei tirocinanti

Postcondizioni: l'azienda non è più convenzionata

Problemi aperti: non permettere all'

Requisiti soddisfatti: 4, 5

Diagramma dei casi d'uso: figura 4.10

Diagramma di sequenza: figura 4.12

Diagramma di sequenza: l'azienda che vuole disdire la convenzione invia la richiesta al sistema, che controlla che non ci siano tirocinanti che stanno svolgendo il tirocinio nell'azienda che ha richiesto la disdetta, poi controlla che non ci siano richieste di tirocinio; se entrambe le condizioni vengono soddisfatte, il sistema elimina l'azienda e con lei elimina tutti i progetti formativi e le proposte che aveva fatto l'azienda.

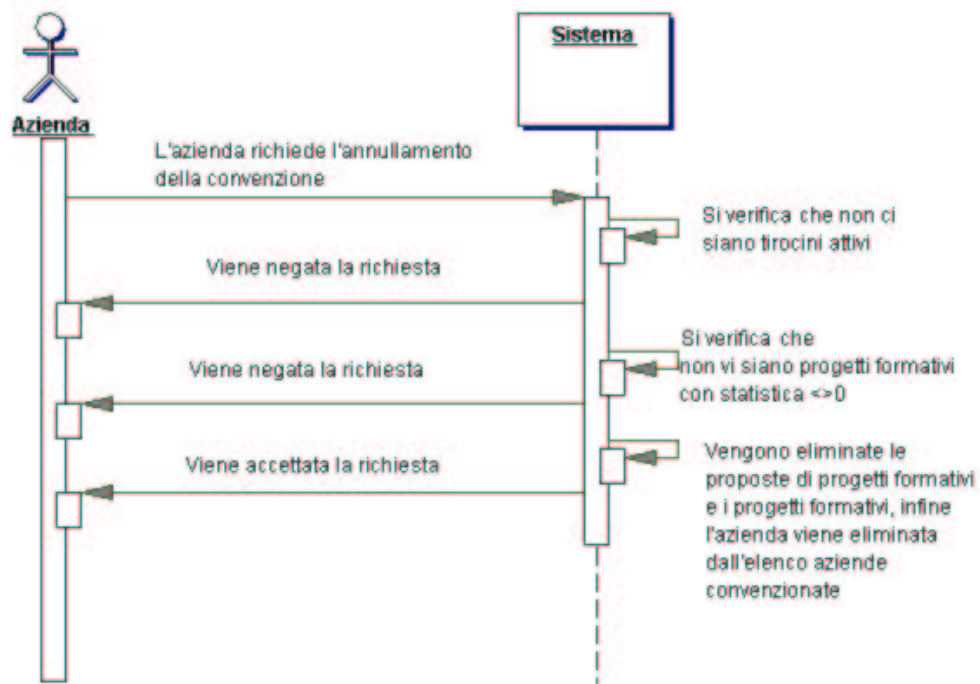


Figura 1.12: disdetta della convenzione

10. Proposta di un progetto formativo

Dichiarazione degli obiettivi: permettere a un'azienda convenzionata di proporre un progetto formativo al Consiglio di corso di studi.

Descrizione:

- a. l'azienda si identifica
- b. l'azienda compila la proposta di progetto formativo
- c. il sistema inserisce nell'elenco la proposta

Priorità: alta

Ipotesi: il sistema ha un elenco dell'aziende convenzionate e delle proposte di progetti formativi

Precondizioni: l'azienda può seguire uno studente in un progetto

Postcondizioni: l'azienda aspetta una risposta da parte del Consiglio di corso di studi

Problemi aperti:

Requisiti soddisfatti: 5

Diagramma dei casi d'uso: figura 4.13

Diagramma di sequenza: figura 4.14

11. Approvazione della proposta di progetto formativo Dichiarazione degli obiettivi:

il Consiglio di corso di studi accetta o meno le proposte di progetti formativi che gli sono pervenute.

Descrizione:

- a. il Consiglio di corso di studi decide se accettare la proposta
- d. inserisce la proposta nell'elenco dei progetti formativi

Alternativa: *il Consiglio decide di non accettare la proposta*

Al passo a. il Consiglio di corso di studi decide di non accettare la proposta. Viene avvertita l'azienda e viene spiegato il motivo del rifiuto.

Priorità: alta

Ipotesi: il sistema ha un elenco delle proposte dei progetti formativi

Precondizioni: il Consiglio di corso di studi è pronto per valutare le proposte

Postcondizioni: sono inseriti nuovi progetti formativi nell'elenco

Problemi aperti:

Requisiti soddisfatti: 5

Diagramma dei casi d'uso: figura 4.13

Diagramma di sequenza: figura 4.14

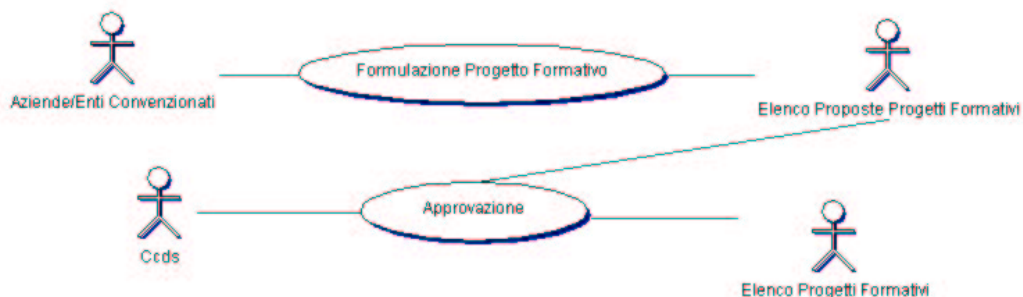


Figura 1.13: proposta e approvazione progetto formativo

Diagramma di sequenza: l'azienda, utilizzando il codice di convenzione, invia al sistema la proposta di progetto formativo. Il sistema inserisce nell'elenco delle proposte il progetto formativo; il consiglio di corso di studi controlla l'elenco; se accetta, la proposta viene inserita nell'elenco progetti formativi, altrimenti viene avvertita l'azienda dei motivi per cui non è stata accettata la proposta.

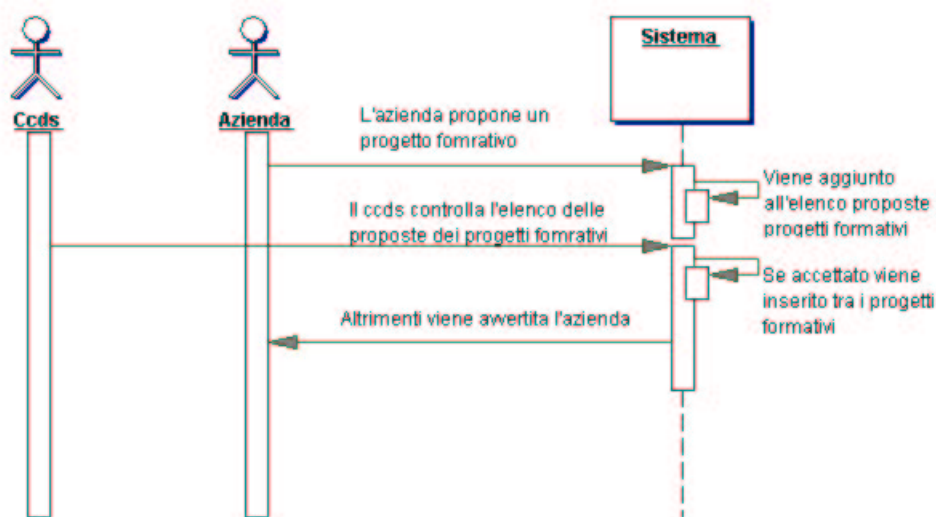


Figura 1.14: proposta e approvazione dei progetti formativi

12. Inserimento di un titolo di prova finale

Dichiarazione degli obiettivi: un docente propone i titoli delle prove finali che ha a disposizione

Descrizione:

- il docente ha un nuovo titolo da proporre
- il docente si identifica al sistema
- il docente inserisce il titolo della prova finale
- il sistema controlla che tutti i campi obbligatori siano compilati

Alternativa: *il docente non viene riconosciuto*

Al passo b. il docente non viene riconosciuto.

Il sistema gli richiede di identificarsi.

Alternativa: *il docente non ha inserito tutti i dati obbligatori*

Al passo d. il docente non ha inserito tutti i dati.

Il sistema dà la possibilità di inserire i dati mancanti.

Priorità: media

Ipotesi: il sistema ha un elenco dei titoli delle prove finali e dei docenti

Precondizioni: il docente ha intenzione di inserire un titolo

Postcondizioni: il titolo è stato inserito

Problemi aperti:

Requisiti soddisfatti: 3

Diagramma dei casi d'uso: figura 4.15

Diagramma di sequenza: figura 4.16

13. Eliminazione dall'elenco dei titoli di prova finale

Dichiarazione degli obiettivi: un docente vuole eliminare un titolo dall'elenco

Descrizione:

- a. il docente si identifica
- b. il docente indica il codice del titolo che vuole eliminare
- c. il sistema controlla il titolo
- d. il sistema elimina il titolo

Alternativa: *il docente non viene riconosciuto*

Al passo a. il docente non viene riconosciuto.

Il sistema gli richiede di identificarsi.

Alternativa: *il sistema non riconosce che il titolo sia del docente che ne richiede l'eliminazione*

Al passo c. il sistema non riconosce la proprietà del titolo.

Il sistema gli nega la possibilità di eliminare il titolo.

Priorità: media

Ipotesi: il sistema ha un elenco dei titoli delle prove finali e dei docenti

Precondizioni: il docente ha intenzione di eliminare un titolo e ha almeno un titolo inserito

Postcondizioni: il titolo è stato eliminato

Problemi aperti:

Requisiti soddisfatti: 3

Diagramma dei casi d'uso: figura 4.15

Diagramma di sequenza: figura 4.16

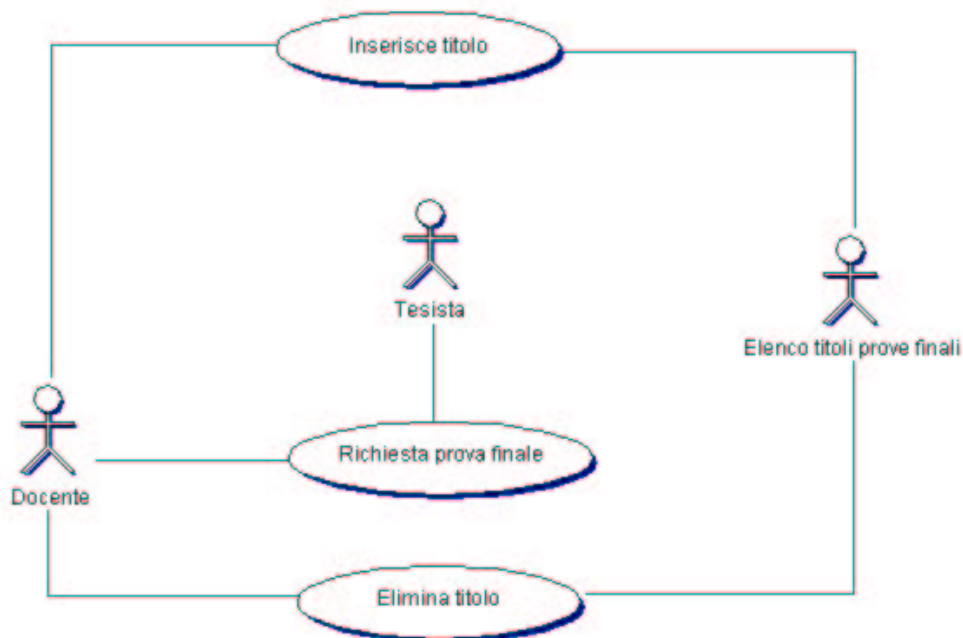


Figura 1.15: inserimento, eliminazione e richiesta della prova finale

Il diagramma di sequenza prevede una prima fase di identificazione del docente, in seguito il docente può introdurre i dati essenziali per la descrizione della tesi.

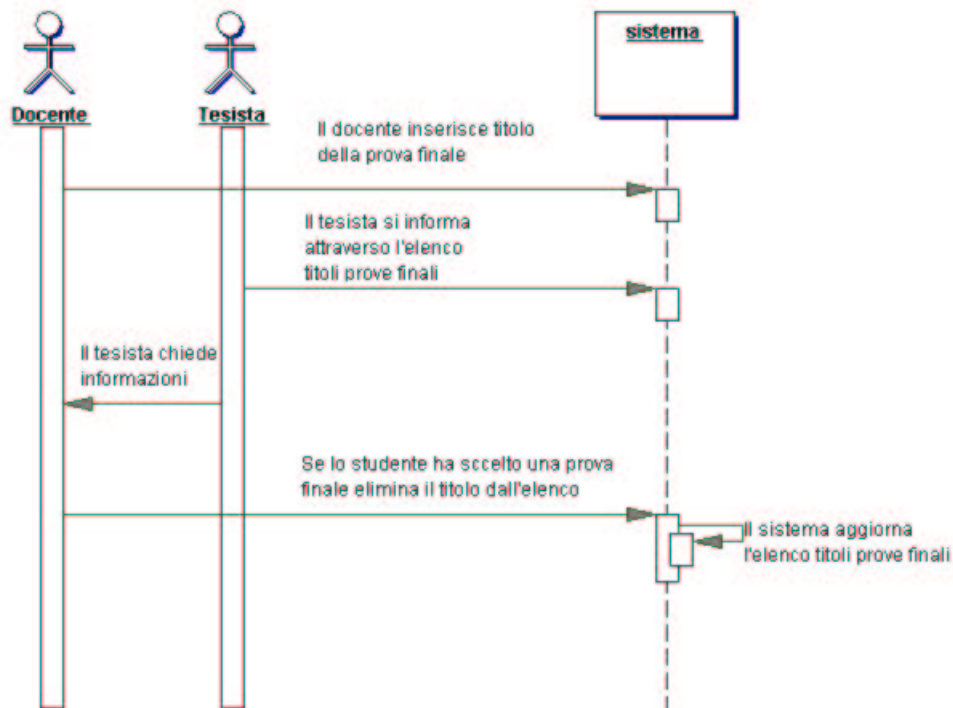


Figura 1.16: inserimento eliminazione e richiesta di una prova finale

Per concludere questa fase, si possono riunire i diagrammi di casi d'uso riguardanti la gestione delle prove finali e dei tirocini in un unico caso d'uso. Questo raggruppamento equivale alla definizione dei package dei casi d'uso. Quindi i package risultano essere:

- Piano di studi: figura 4.3
- Tesi: figura 4.17
- Tirocinio: figura 4.18

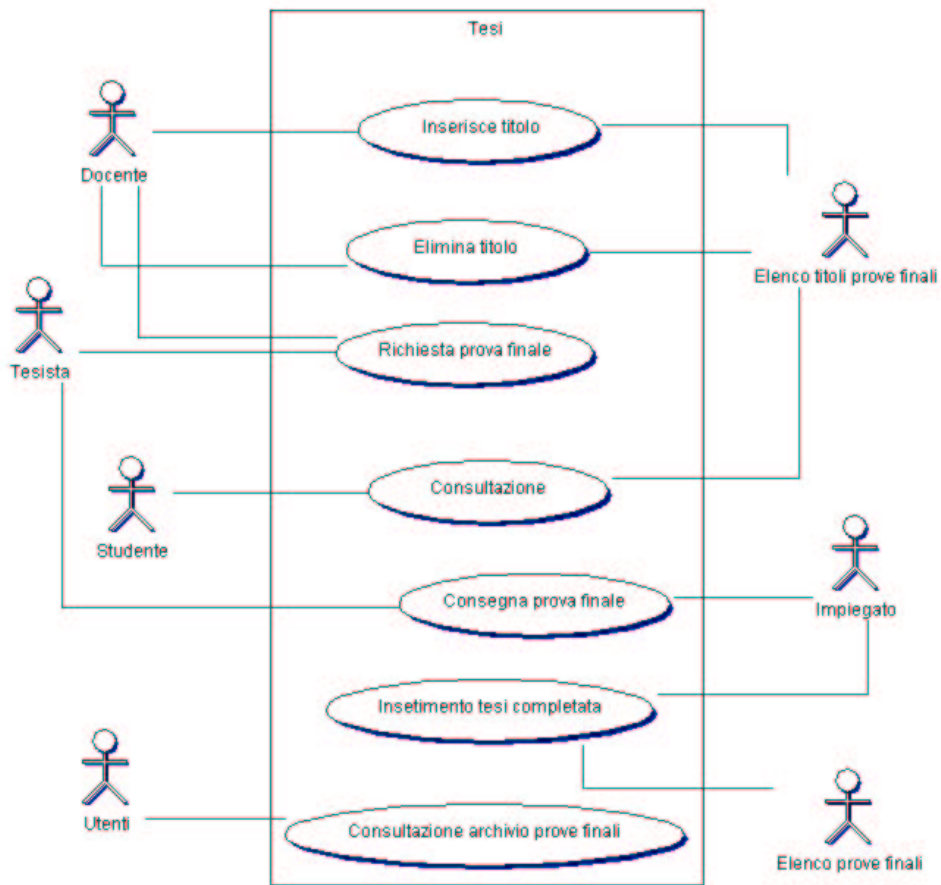


Figura 1.17: prova finale

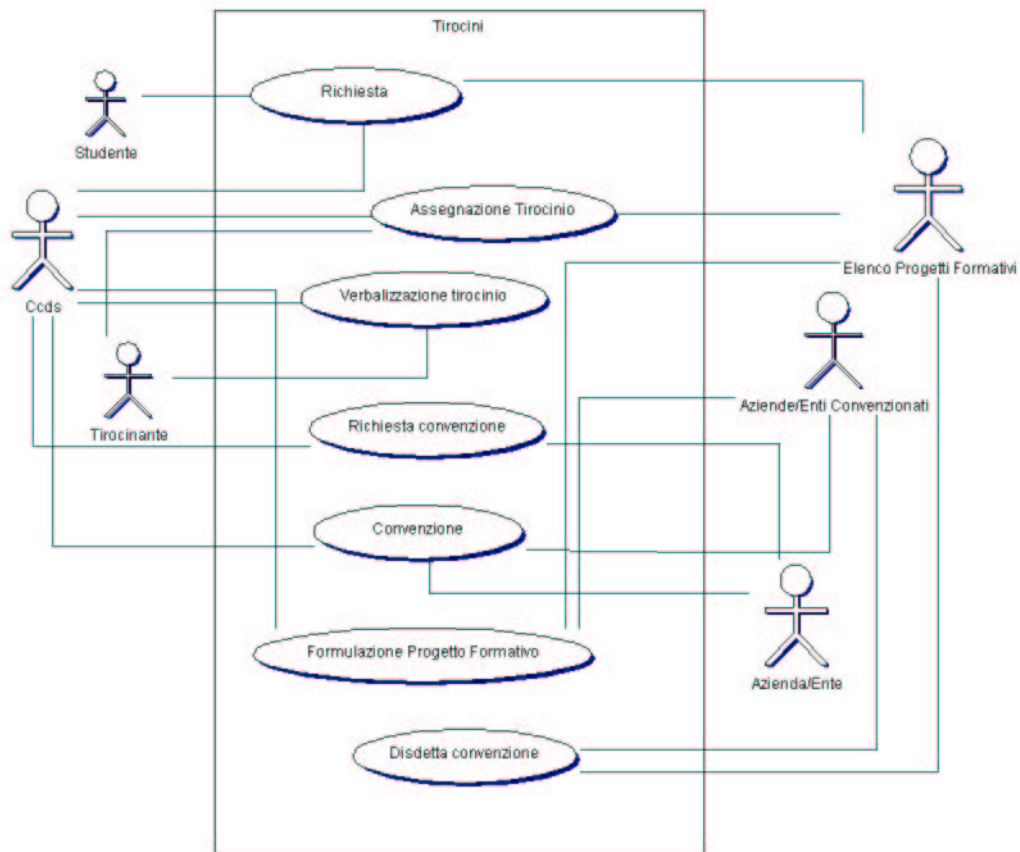


Figura 1.18: tirocinio

1.2 Conclusioni

In questo capitolo si è sviluppata tutta la prima fase del ciclo di vita di un'applicazione Web, l'analisi dei requisiti. Ora si può passare alla fase di analisi e progettazione avendo chiari gli obiettivi e le funzioni del nostro sistema. Nella progettazione si svilupperanno i requisiti funzionali del sistema.

Capitolo 2

Progettazione di un'applicazione per servizi universitari

2.1 Introduzione

Questo capitolo prende in esame innanzitutto l'analisi dei casi d'uso, per rappresentare il diagramma dei package. Dentro a ogni package sono racchiuse delle classi, classi che verranno studiate una per una. Infine vengono presentati i diagrammi di iterazione, in particolare i diagrammi di sequenza e quelli di attività.

2.2 Analisi dei casi d'uso

Si rappresentano gli oggetti: limite con un'ellisse, entità con un cerchio, controllo con un attore inscritto in un cerchio.

1. Consegna piani di studi

Questo caso d'uso prevede la redazione del piano di studi via Web. Sono stati previsti due diversi oggetti limite, il primo per l'identificazione dello studente, il secondo per la compilazione del piano di studi. In seguito l'oggetto controllo inserisce il piano di studi nell'entità elenco piani di studi da approvare.

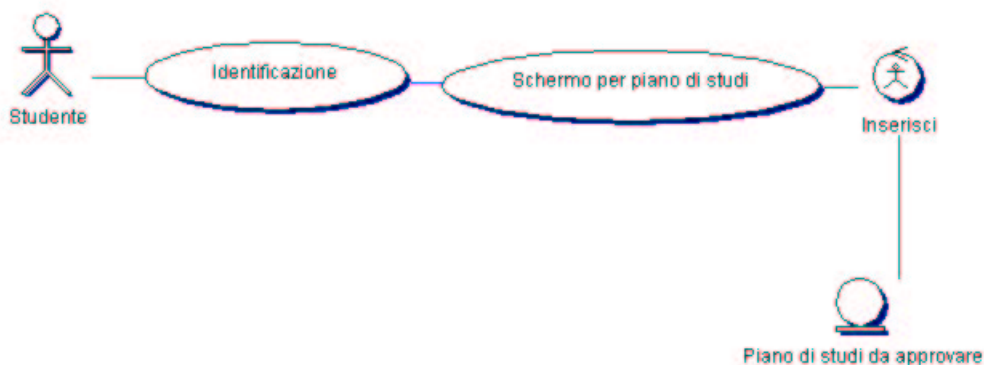


Figura 2.1: consegna piano di studi

2. Approvazione piani di studi

Il Consiglio di corso di studi (Ccds), accede all'oggetto limite dove sono inseriti l'elenco piano di studi da approvare e le carriere degli studenti. Se il Consiglio approva, il piano di studi è approvato e vengono inseriti in carriera tutti gli esami che lo studente deve ancora sostenere, altrimenti lo studente viene avvertito.

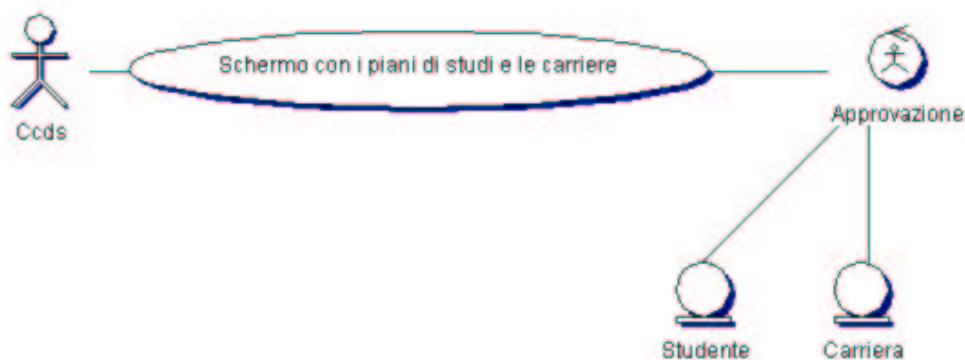


Figura 2.2: Ccds approva il piano di studi

3. Richiesta tirocinio

Il tirocinante, dopo che si è identificato, richiede attraverso l'oggetto limite il tirocinio che vorrebbe svolgere. Vengono richiamati due oggetti controllo per l'inserimento della richiesta nell'elenco domande di tirocinio e per la modifica delle statistiche nell'elenco progetti formativi. Questa statistica viene

aggiornata per mantenere in memoria il numero di richieste totali per ciascuna proposta di tirocinio.



Figura 2.3: richiesta di tirocinio

4. Assegnazione tirocinio Il Consiglio di corso di studi (Ccds), dopo essersi identificato, accede all'oggetto limite dove sono inserite le domande di tirocinio, decide come assegnare i tirocini e li inserisce nell'elenco.

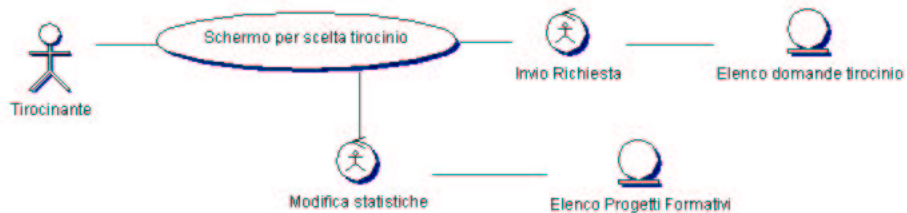


Figura 2.4: assegnazione tirocini

5. Scelta prova finale

In questo caso non viene fatta nessuna analisi poiché lo studente che accede all'elenco non può modificare in nessun modo il sistema.

6. Consegna della prova finale

Il diagramma illustra l'inserimento della tesi attraverso un oggetto limite dove vengono indicati le parti che possono aiutare nell'archiviazione della tesi di laurea (metadati), e la tesi stessa. Il sistema, attraverso un oggetto controllo, controlla i metadati e in seguito gli stessi vengono inseriti nel sistema.

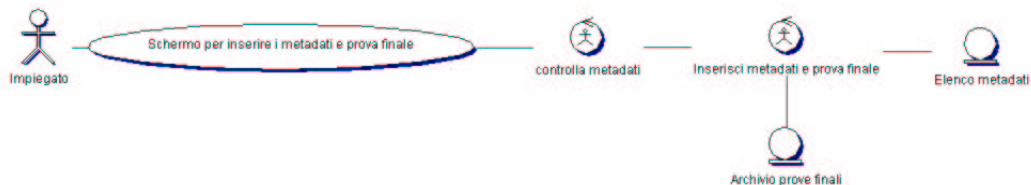


Figura 2.5: inserimento prova finale

7. Richiesta convenzioni, 9. Disdetta convenzione e 10. Proposta progetto formativo

Si prevede che l'azienda possa essere già convenzionata oppure voglia richiedere la convenzione.

Nel caso in cui l'azienda sia già convenzionata avrà un oggetto limite che le chiederà se vuole inserire un nuovo progetto formativo oppure se desidera rinunciare alla convenzione.

Azienda che vuole inserire una nuova proposta di progetto formativo: attraverso un oggetto di controllo "inserisce proposta progetto formativo" viene modificato l'elenco delle proposte dei progetti formativi.

Azienda che vuole disdire la convenzione: attraverso l'oggetto di controllo "controlla" vengono controllati i progetti formativi attivi e l'elenco dei progetti formativi. Se il controllo è positivo, il sistema elimina l'azienda dall'elenco aziende convenzionate, tutte le proposte di progetto formativo e i progetti formativi approvati.

Se un'azienda si vuole convenzionare, vengono inseriti i suoi dati nell'elenco richieste convenzioni.

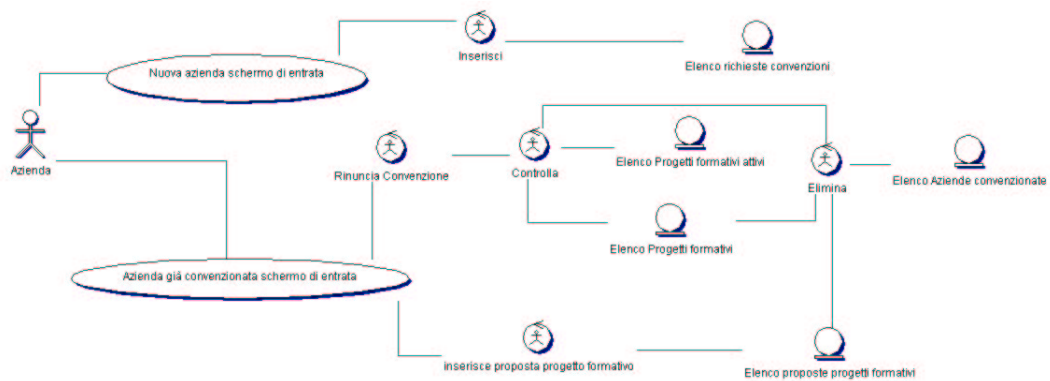


Figura 2.6: gestione convenzioni aziende

8. Approvazione della convenzione

Il Consiglio di corso di studi accede all'elenco delle richieste. Le aziende che sono approvate attraverso l'oggetto di controllo approvazione vengono inserite nell'elenco aziende convenzionate, le altre vengono avvertite del rifiuto.

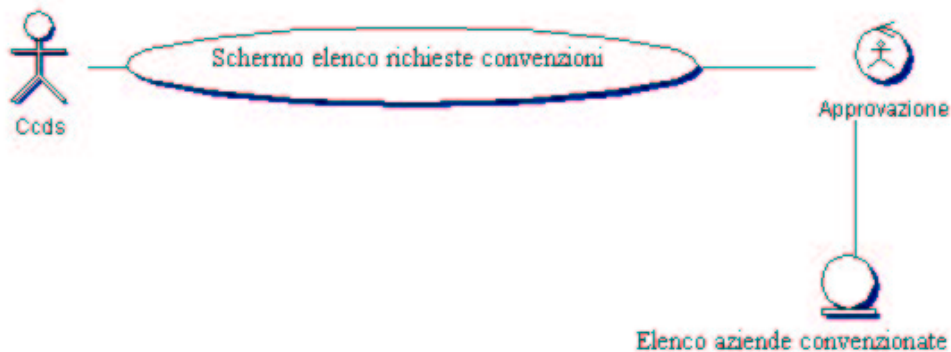


Figura 2.7: approvazione convenzione

11. Approvazione proposte progetti formativi

Il Consiglio di corso di studi accede all'elenco delle proposte progetti formativi. I progetti che vengono approvati sono inseriti nell'elenco progetti formativi. Le aziende dei progetti formativi non approvati vengono contattate.

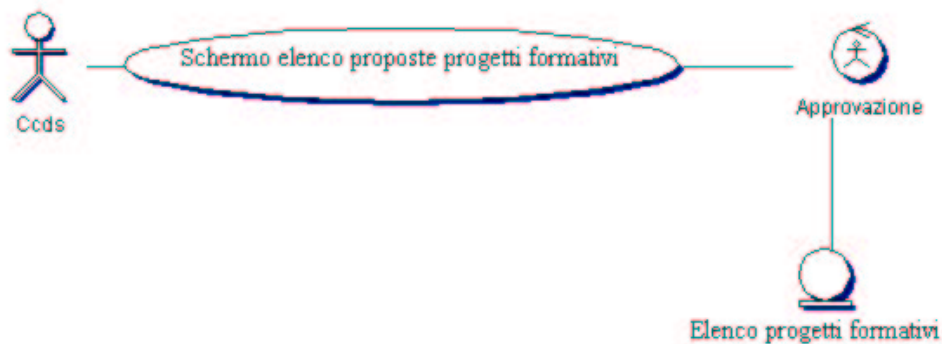


Figura 2.8: approvazione proposta progetto formativo

12. Inserimento ed eliminazione di un titolo

Il docente accede a due oggetti limite in cui può inserire o eliminare il titolo di una prova finale. Questo provoca una modifica dell'oggetto entità elenco titoli prova finale.

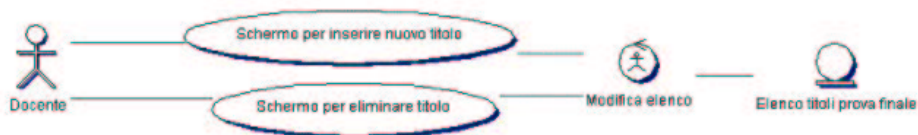


Figura 2.9: inserimento ed eliminazione titoli prova finale

2.3 Progettazione

Il progetto viene rappresentato come un insieme di sottoproblemi più facili da gestire, rappresentati in UML come package. In ogni package sono contenute le classi che servono per la gestione dello stesso. Il diagramma dei package è riportato nella figura 5.10. I package principali sono: Piano di studi, Prova finale, Tirocini.

I package di Prova finale e Tirocini dipendono da Piano di studi: infatti uno studente può iniziare a lavorare alla prova finale e a svolgere il tirocinio solo nel momento in cui la sua carriera glielo permette.

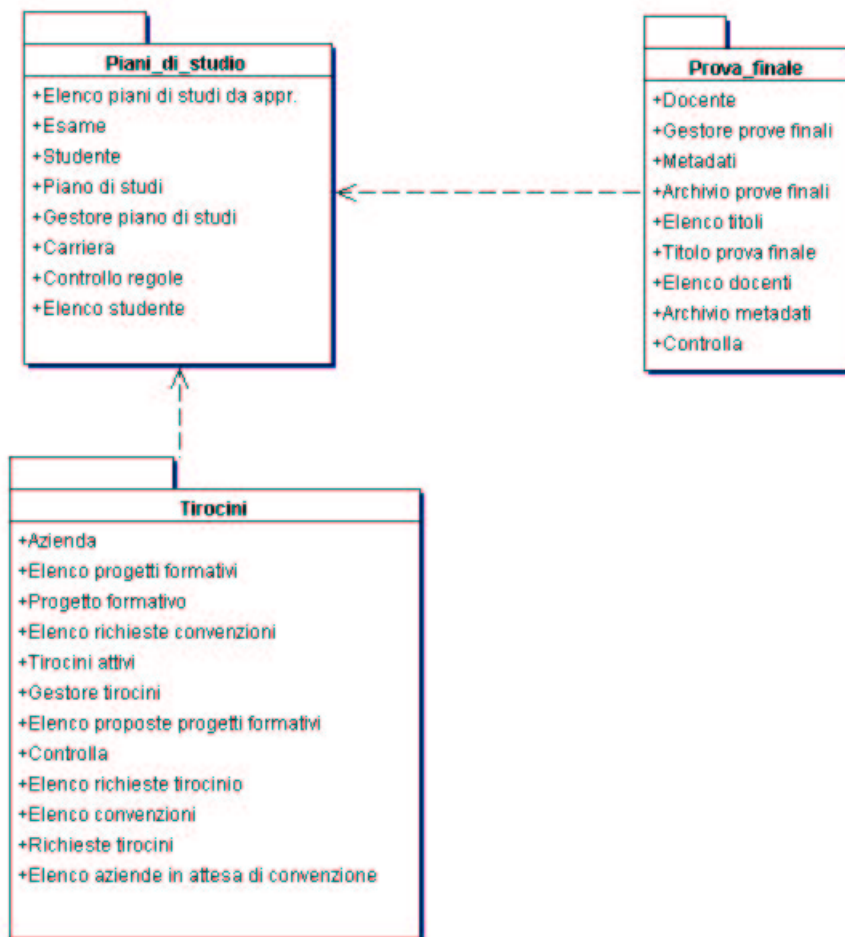


Figura 2.10: package

Qui di seguito vengono presentati i package e per ciascuno vengono illustrati i diagrammi delle classi, di sequenza e delle attività.

2.3.1 Piano di studi

Nel diagramma delle classi sono rappresentate le seguenti classi: Esame, Studente, Elenco_studenti, Piano_di_studi, Elenco_piani_di_studi_da_approvare, Carriera, Controllo_regole, Gestore_piani_di_studi.

L'obiettivo principale di questo package è la gestione completa del servizio

che l'applicazione Web dovrebbe fornire agli studenti per la compilazione e l'approvazione dei piani di studio.

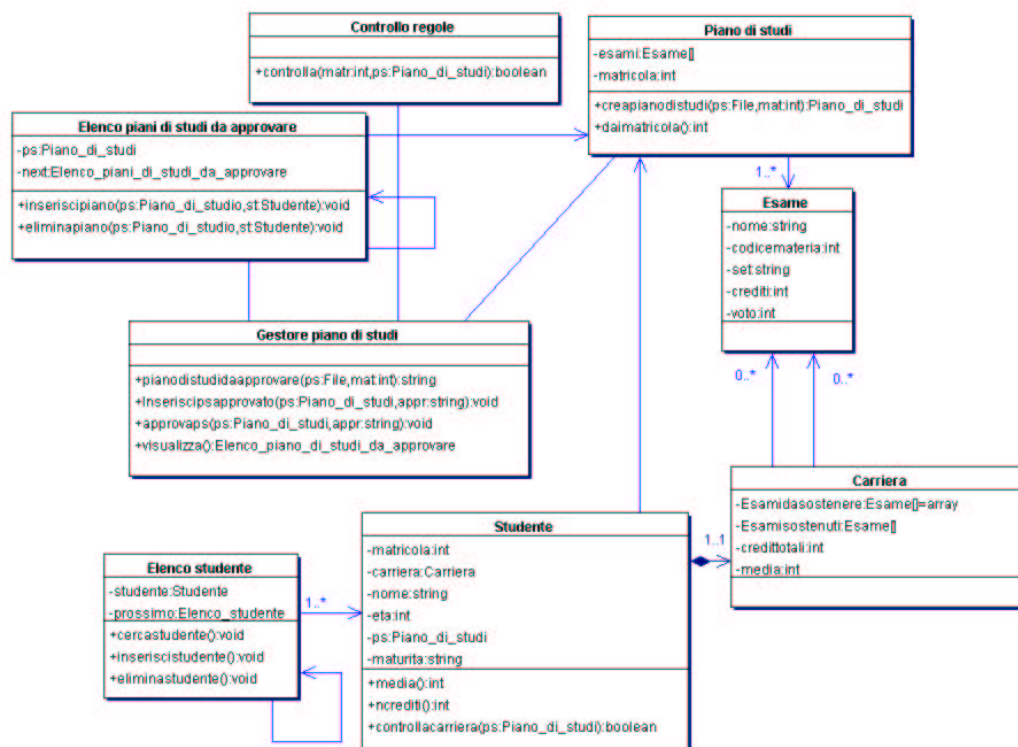


Figura 2.11: package piano di studi

Descrizione dei compiti delle classi e dei loro metodi.

Gestore_piano_di_studi

Descrizione

Questa classe è la più importante del package e rappresenta il punto di ingresso per le operazioni compiute al suo interno.

Operazioni

Inserisci piano di studi da approvare(): prende in input un file, e un intero che rappresenta il numero di matricola dello studente, per prima cosa viene richiamato il metodo controlla della classe Controllo_regole, questo metodo prende in input il numero di matricola, e dice se lo studente può fare la domanda per il piano di studi, l'operazione torna un valore booleano, se questo è "true" allora inserisce il piano di studi attraverso il metodo creapianodistudi(),

creastudente() e inseriscipiano(), altrimenti il sistema restituisce un segnale di errore allo studente.

approvaps(): prende in input un oggetto del tipo Piano_di_studi, un valore string in cui viene indicato se il piano di studi è approvato, e un oggetto di tipo Studente. Questo metodo richiama il metodo inseriscipinaodistudi() che inserisce in carriera gli esami che sono elencati nel piano di studi. Se non viene approvato il piano di studi viene avvertito lo studente, infine viene richiamato il metodo eliminapiano() della classe Elenco_piano_di_studi_da_approvare.

visualizza(): restituisce l'elenco dei piani di studi da approvare.

Studente

Descrizione

La classe Studente viene definita dagli attributi che seguono: matricola, carriera, nome, cognome, eta, maturita e ps (piano di studi).

Operazioni

media(): valori di ritorno la media dello studente

ncrediti(): valori di ritorno il numero di crediti dello studente

controllacarriera(): controlla se ha sostenuto gli esami che richiede nel piano di studi

Elenco_studenti

Descrizione

Rappresenta tutti gli studenti presenti nel corso di studi

Operazioni

inserisci(): prende in input un oggetto di tipo studente e lo inserisce nell'elenco

elimina(): prende in input un oggetto di tipo studente e lo elimina

cerca(): prende in input il numero di matricola e restituisce lo studente associato.

Elenco_piani_di_studi_da_approvare

Descrizione

È l'insieme di tutte le richieste di piani di studi degli studenti

Operazioni

inseriscipiano(): inserisce il piano di studi che prende in input

eliminapiano(): elimina il piano di studi che prende in input

Carriera

Descrizione

In carriera sono elencati tutti gli esami che lo studente ha sostenuto, con i relativi voti, i crediti totali e la media.

Operazioni

inserisciesamedasostenere(): inserisce gli esami che lo studente deve sostenere per laurearsi

inserisciesamesostenuto(): elimina l'esame che prende in input dall'array esami da sostenere e inserisce l'esame in esame sostenuti.

Controllo_regole

Descrizione

Controllo_regole serve per verificare che lo studente può richiedere il piano di studi, e che gli esami inseriti nel piano di studi non siano già nella carriera dello studente

Operazioni

Controlla(): prende in input la matricola e il piano di studi, vengono fatti i controlli sullo studente e sugli esami che ha in carriera.

Piano_di_studi

Descrizione

Definisce un piano di studi, che non è altro che un array di esami collegati a un determinato studente.

Operazioni

creapianodistudi(): è il metodo della classe che crea un oggetto di tipo Piano_di_studi

Esame

Descrizione

Gli attributi che definiscono un Esame sono: nome, codicemateria, set (il settore disciplinare della materia), crediti (quanti crediti vale l'esame), voto.

Diagramma di sequenza

Richiesta piano di studio

Nel diagramma vengono mostrati come il sistema e lo studente interagis-

cono. Lo studente attraverso il metodo `pianodistudidaapprovare()` richiede l'inserimento del suo piano di studi; il `Gestore_piano_di_studi()` richiama la funzione `controlla`, a sua volta invoca la funzione `controllacarriera()` che controlla che lo studente possa richiedere il piano di studi; se quest'ultimo ritorna un valore `true`, il `Gestore_piano di studi()` inserisce il piano di studi nell'elenco altrimenti, termina con un messaggio di errore.

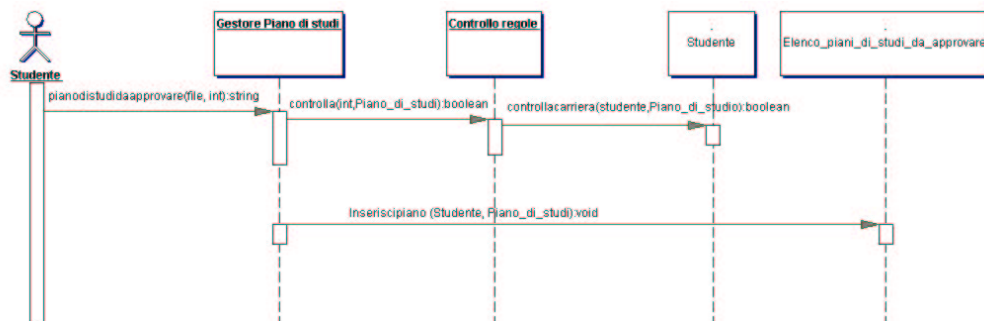


Figura 2.12: richiesta piano di studi

Approvazione piano di studio

Nel diagramma seguente viene illustrato come il Consiglio di corso di studi interagisce con i vari oggetti del sistema.

Il Consiglio chiede al `Gestore_pinao_di_studi()` di visualizzare l'elenco dei piani di studi da approvare. Per ciascun piano di studi il Consiglio di corso di studi richiama il metodo `approva piano di studi`. Se lo approva vengono inseriti gli esami in carriera, altrimenti viene avvertito lo studente. Comunque sia viene eliminato il piano di studi dall'elenco piano di studi da approvare con il metodo `eliminapiano()`

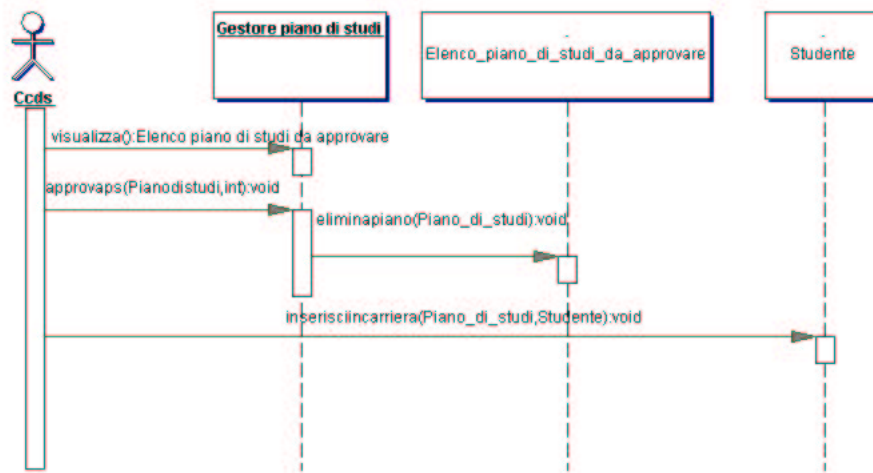


Figura 2.13: approvazione piano di studi

Diagramma delle attività

Approvazione piano di studio

Nel diagramma che segue vengono indicate quali sono le attività per approvare il piano di studi. Lo studente, se viene riconosciuto, può compilare il suo piano di studi, quindi viene controllata la correttezza del piano di studi, se risulta corretto viene inserito nell'elenco piani di studi da approvare. A questo punto, se viene approvato, viene aggiornata la carriera dello studente, altrimenti viene avvertito lo studente.

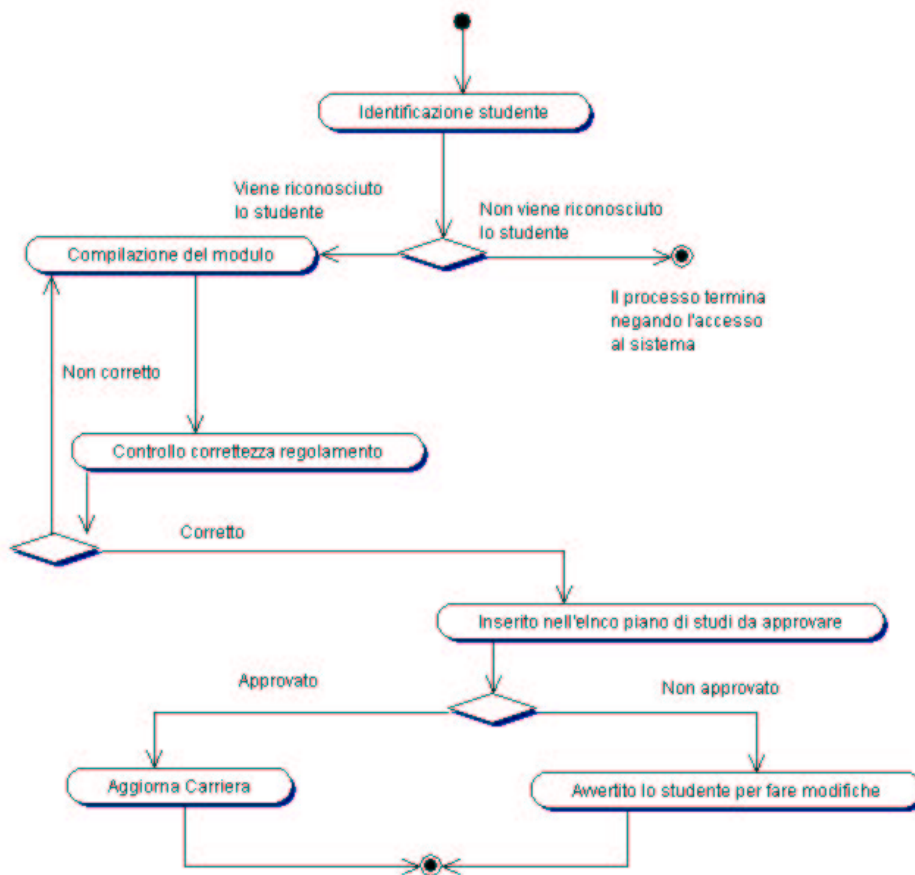


Figura 2.14: approvazione piano di studi

2.3.2 Tirocini

Nel package dei tirocini sono contenute tutte le classi che vengono usate per la gestione delle convenzioni, delle proposte di progetti formativi e delle domande di tirocinio. La classi contenute in questo diagramma sono:

Gestore_tirocini, Elenco_convenzioni, Elenco_richieste_convenzioni, Aziende, Elenco_progetto_formativo, Progetto_formativo, Elenco_proposte_
 _progetti_formativi, Tirocini_attivi, Controlla, Richieste_tirocini ed Elen-
 co_richieste_tirocini.

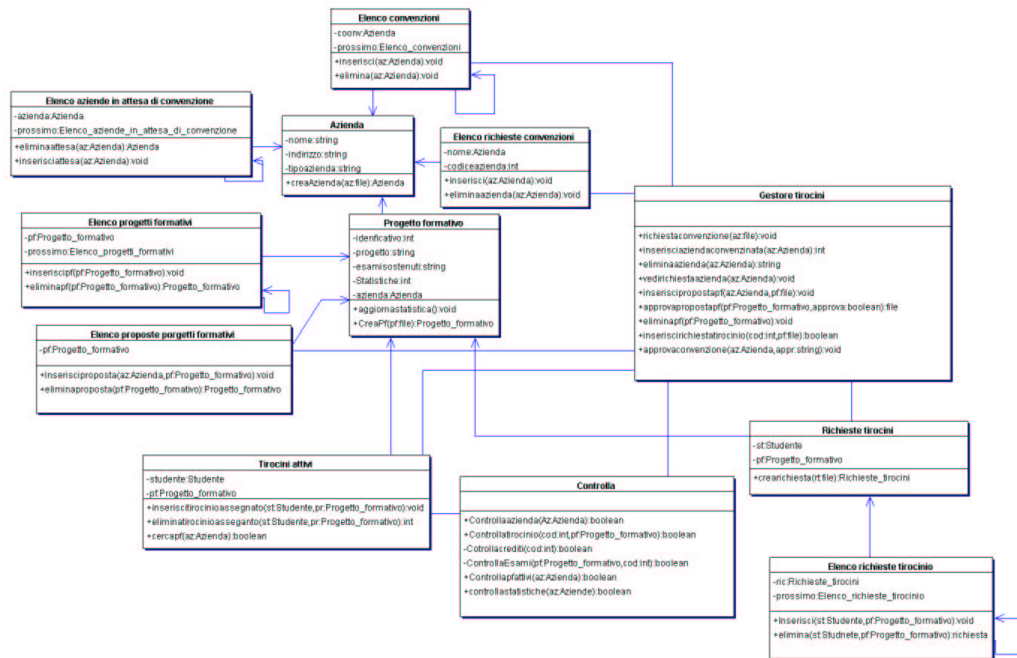


Figura 2.15: package tirocini

Descrizione dei compiti delle classi e dei loro metodi.

Gestore_tirocini

Descrizione

Il gestore dei tirocini è la classe che gestisce tutte le classi del package.

Operazioni

richiestaconvenzione(): inserisce un'azienda nell'elenco richieste convenzioni utilizzando il metodo di Azienda creaazienda() e il metodo di Elenco_richieste_convenzioni inserisci()

approvaconvenzione(): approva le convenzioni, inserisce un'azienda nell'elenco delle aziende in attesa di convenzione attraverso il metodo inserisciattesa() della classe Elenco_aziende_in_attesa_di_convenzione ed elimina l'azienda dall'elenco richiesta convenzione sia se è stata accettata la convenzione sia in caso contrario attraverso il metodo eliminaazienda() della classe Elenco_richieste_convenzioni.

inserisciaziendaconvenzionata(): questo metodo inserisce un oggetto della classe Azienda nell'Elenco_convenzioni, invocando il metodo eliminaatte-

sa() della classe Elenco_aziende_in_attesa_di_convenzione e inserisci() della classe Elenco_convenzioni.

vedirichiestaazienda() visualizza l'elenco richieste convenzioni.

eliminaazienda() viene chiamata questa funzione per eliminare un'azienda dall'elenco convenzioni. Il metodo prima controlla attraverso il metodo controllaazienda() della classe Controlla se non ci sono tirocini attivi e non ci sono statistiche diverse da zero. Se il valore del controllo è true allora l'azienda viene eliminata altrimenti viene ritornato un messaggio di errore.

inseriscipropostapf(): si inserisce una proposta di progetto formativo nell'elenco.

approvapropostapf(): vengono approvate le proposte di progetto formativo, il metodo richiama il metodo della classe Elenco_proposte_progetti_formativi elimina proposta() e il metodo inseriscipf() della classe Elenco_progetti_formativi.

inseriscirichiestatirocinio(): inserisce una richiesta in Richieste_tirocini, dopo aver controllato con il metodo controllatirocinio() della classe Controlla.

assegnazionetirocini() assegna i tirocini, invoca il metodo elimina() della classe Elenco_progetto_formativo ed infine inserisce i dati in Tirocini_attivi attraverso il metodo inseriscitirocinioassegnato().

Elenco_richieste_tirocini

Descrizione

Vengono inserite tutte le domande di tirocinio.

Operazioni

Inserisci() inserisce una nuova proposta.

Elimina() elimina una proposta.

Controlla

Descrizione

Vengono effettuati i diversi controlli di correttezza.

Operazioni

Controllaaziendacon(): controlla che l'azienda non abbia dei tirocini attivi o delle richieste per fare un tirocinio.

Controllatirocinio(): controlla che lo studente possa richiedere il tirocinio e che per svolgere il tirocinio abbia svolto gli esami necessari.

Controllapfattivi(): cerca se ci sono dei tirocini attivi presso un'azienda.

ControllaEsami(): controlla se gli esami richiesti dal tirocinio sono stati sostenuti dallo studente.

Controllacrediti():controlla che lo studente abbia i crediti necessari per fare il tirocinio.

Controllastatistiche(): controlla che le statistiche siano diverse da zero.

Elenco _proposte _progetti _formativi

Descrizione

Vengono inserite tutte le proposte che le aziende fanno per un progetto formativo.

Operazioni

inserisciproposta(): inserisce una proposta di tirocinio nell'elenco.

eliminaproposta() :inseriscono ed eliminano un elemento dall'elenco.

Tirocini _attivi

Descrizione

Vengono indicati tutti i tirocini che sono svolti dagli studenti.

Operazioni

inseriscitirocinioassegnato(), eliminatirocinioassegnato(), cercapf(): vengono inseriti,eliminati e cercati i tirocini attivi.

Richiesta _tirocini

Descrizione

Vengono definiti da questa classe gli oggetti di tipo Richiesta _tirocini che hanno come attributi: Studente e Progetto _formativo.

Operazioni

crearichiesta(): dato un file ritorna un oggetto.

Elenco _richieste _convenzioni

Descrizione

Vengono inserite tutte le richieste di tirocini.

Operazioni

inserisciazienda():inserisce un'azienda

eliminaazienda():elimina un'azienda

Elenco _ convenzioni

Descrizione

Sono inserite tutte le aziende convenzionate.

Operazioni

inserisciconv(): inserisce un oggetto Azienda in Elenco _ convenzioni.

eliminaconv(): richiama i metodi elimina() della classe Elenco _ progetti _ formativi ed elimina() della classe Elenco _ proposte _ progetti _ formativi.

Aziende

Descrizione

Vengono definite le aziende.

Operazioni

creaazienda(): si crea un oggetto di tipo azienda.

Progetto _ formativo

Descrizione

Viene definito un oggetto Progetto _ formativo.

Operazioni

aggiornastatistiche (): ogni volta che viene fatta una richiesta di un progetto formativo, viene aggiornata la statistica relativa.

creapf(): viene creato un oggetto appartenente alla classe.

Elenco _ richieste _ convenzioni

Descrizione

Sono inserite tutte le aziende che richiedono di essere convenzionate.

Operazioni

inserisciazienda(): prende in input un'azienda e l'inserisce in elenco.

eliminaazienda(): elimina l'azienda dalla lista che gli è stata passata in input.

Elenco _ progetti _ formativi

Descrizione

Vengono inseriti tutti i progetti formativi disponibili.

Operazioni

inseriscipf(): prende l'input da Progetto formativo e lo inserisce in Elenco_progetti_formativi.

eliminapf(): prende in input un Progetto formativo da eliminare dall'elenco.

Elenco_aziende_attesa_di_convenzione

Descrizione

Sono le aziende che sono state accettate dal consiglio di corso di studi, ma non hanno ancora mandato la documentazione.

Operazioni

inserisciattesa(): inserisce un oggetto Azienda nell'Elenco_aziende in _attesa_di_convenzione.

eliminaattesa(): elimina un oggetto Azienda dall'Elenco_aziende in _attesa_di_convenzione.

Diagramma di sequenza Richiesta tirocinio

inseriscirichiestatirocinio(): prende in input il codice dello studente e un file con la richiesta di tirocinio; il Gestore_tirocini controlla se lo studente ha i crediti necessari per effettuare il tirocinio; se ritorna vera la funzione, confronta gli esami del Progetto formativo con gli esami sostenuti dallo studente; se anche questo controllo torna vero, vengono inseriti, con il metodo inserisci() della classe Gestore_tirocini, gli oggetti Studente e Progetto_formativo; il Gestore-tirocini infine aggiorna le statistiche del Progetto formativo.

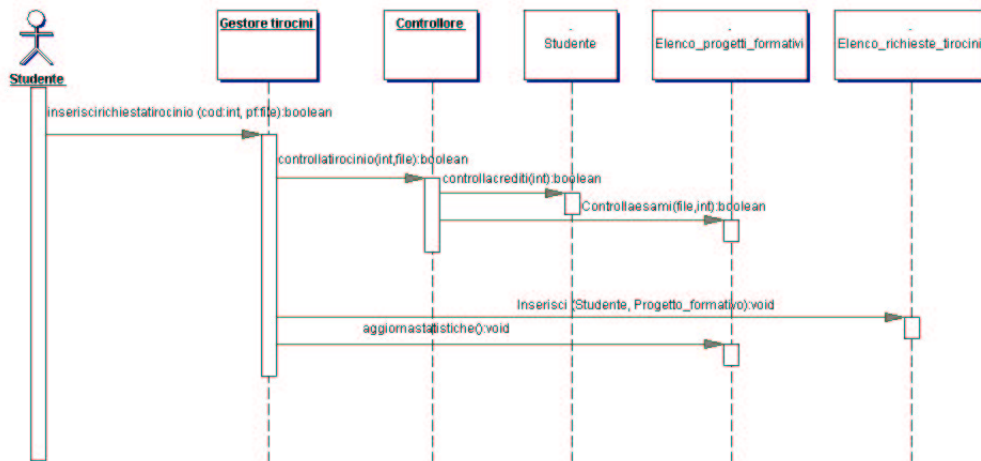


Figura 2.16: richiesta tirocinio

Richiesta di convenzione

L'azienda richiede di convenzionarsi attraverso il metodo `richiestaconvenzione()` che prende in input un file; il `Gestore_tirocini` trasforma il file in un oggetto `Azienda` attraverso il metodo `creaazienda()` della classe `Azienda` e poi inserisce l'azienda nell'elenco `Richieste e Convenzioni`.

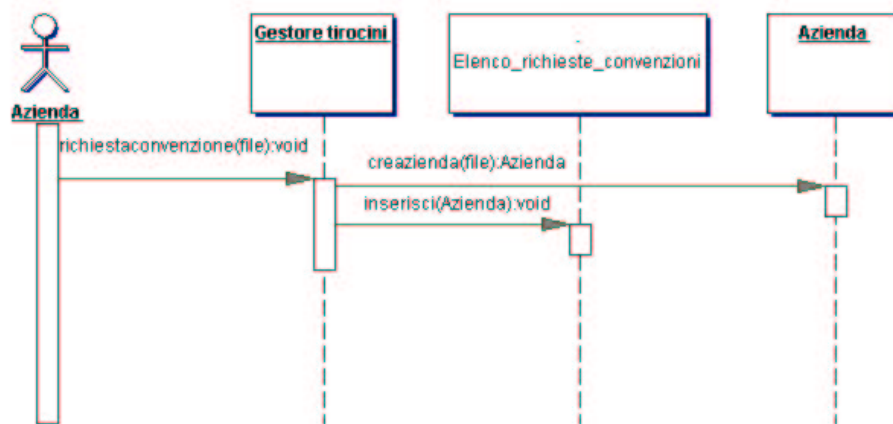


Figura 2.17: richiesta convenzione

Convenzione azienda

Il Consiglio di corso di studi visualizza l'`Elenco_richieste_convenzioni` al `Gestore_tirocini`; decide se convenzionare o meno l'azienda, utilizzando il

metodo `approvaconvenzione()`; il `Gestore_tirocini` invoca il metodo `eliminaazienda()` dall'`Elenco_richieste_convenzioni` e poi il metodo `inserisciaziendainataesa()` nell'`Elenco_aziende_in_attesa_di_convenzione`; il Consiglio di corso di studi aspetta i documenti dell'azienda; una volta ricevuti i documenti, inserisce l'azienda con il metodo `inserisciaziendaconvenzione()` nell'`Elenco_convenzioni`.



Figura 2.18: convenzione azienda

Inserimento nuovo progetto formativo.

L'azienda, con il metodo `inseriscipropostapf()`, chiede al `Gestore_tirocini` di inserire la sua proposta di progetto formativo; il `Gestore_tirocini`, con il metodo `controllaazienda()`, controlla che l'azienda sia presente nell'`Elenco_convenzioni`; se il controllo torna vero, il `Gestore_tirocini`, con il metodo `inserisciproposta()`, inserisce il progetto formativo nell'`Elenco_proposte__progetti_formativi`.

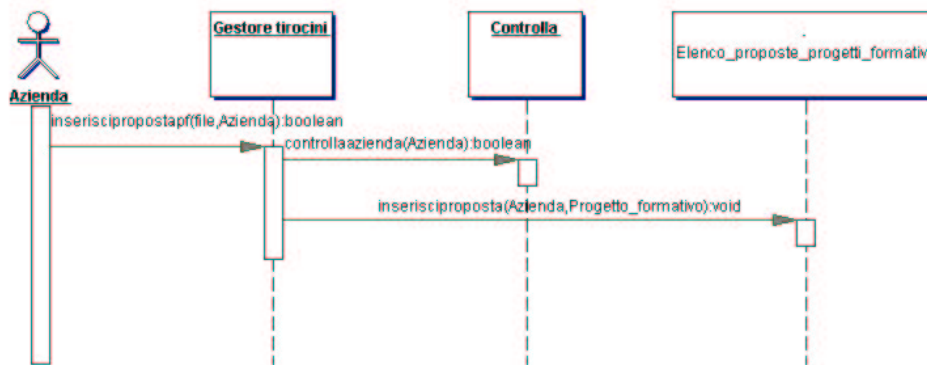


Figura 2.19: nuovo progetto formativo

Disdetta convenzione

L'azienda, con il metodo `eliminaazienda()`, chiede al `Gestore_tirocini` di essere eliminata dall'`Elenco_convenzioni`; il `Gestore_tirocini`, con il metodo `controllaaziendaconv()`, verifica che non ci siano tirocini attivi e che le statistiche dei progetti formativi siano diverse da zero, con i metodi `cercapfativo()` e `controllastatistiche()`; se tornano veri entrambi i valori, il `Gestore_tirocini`, con il metodo `eliminaazienda()`, cancella l'azienda dall'`Elenco_convenzioni`.

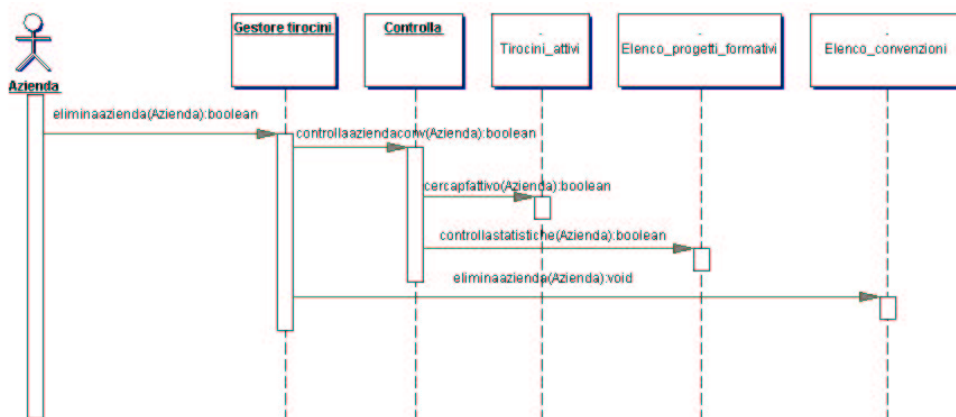


Figura 2.20: disdetta convenzione

Diagramma delle attività

Richiesta convenzione e disdetta convenzione

L'azienda che richiede di essere convenzionata invia la richiesta al Consiglio

di corso di studi attraverso l'applicazione Web; se la richiesta viene approvata, l'azienda, dopo avere inviati i documenti necessari, riceve un codice di convenzione.

L'azienda che richiede la disdetta della convenzione invia la richiesta all'applicazione Web che controlla che non vi siano tirocini attivi o richieste di tirocinio; se entrambi i valori tornano veri, allora l'azienda viene cancellata e, di conseguenza, vengono cancellati anche tutti i progetti formativi a lei associati.

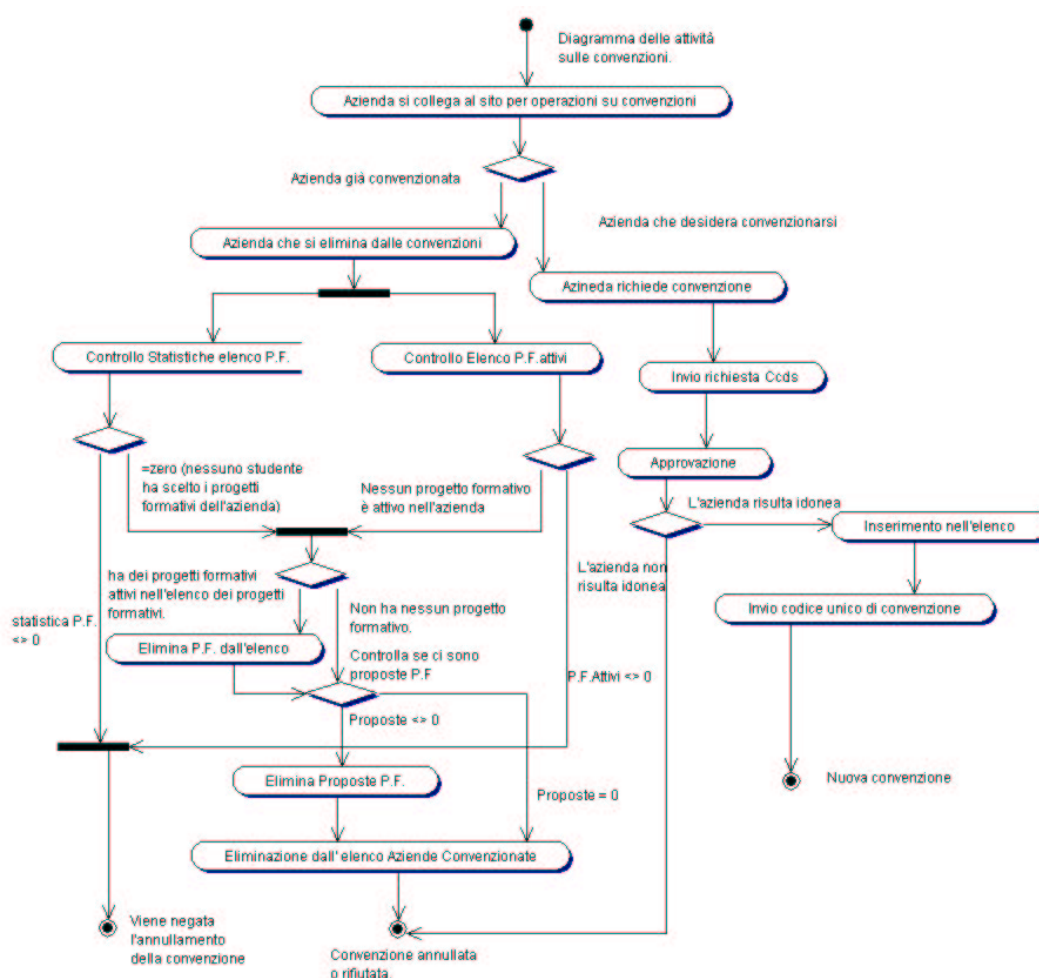


Figura 2.21: richiesta e disdetta convenzione

Proposta e accettazione progetto formativo

L'azienda che richiede l'approvazione di un progetto formativo invia la richi-

esta al Consiglio di corso di studi attraverso l'applicazione Web; l'applicazione Web controlla che l'azienda sia convenzionata e inserisce la proposta nell'elenco proposte progetti formativi. Il Consiglio di corso di studi, se accetta il progetto, lo inserisce nell'elenco progetti formativi.

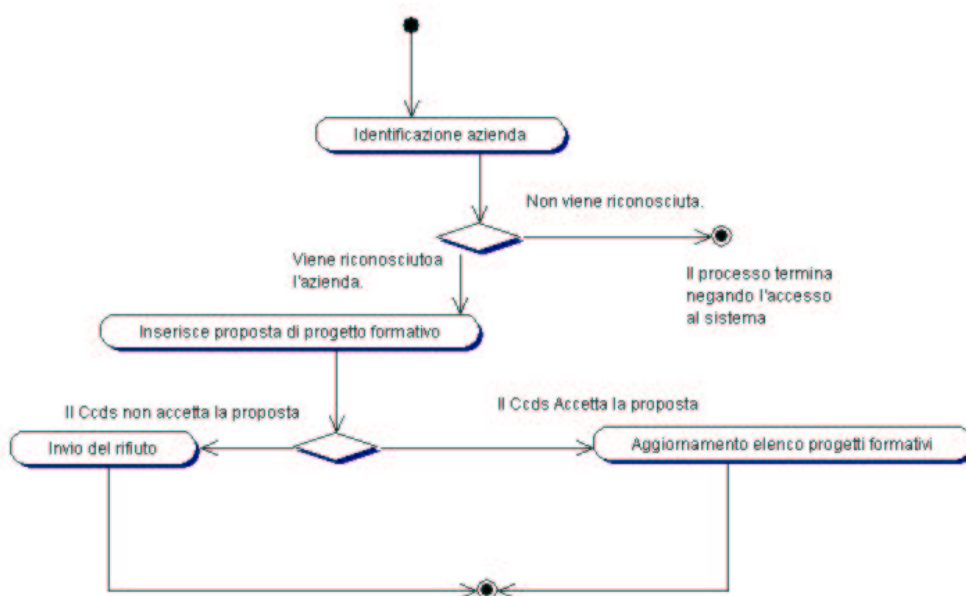


Figura 2.22: proposta e accettazione progetto formativo

Richiesta e assegnazione

Lo studente, dopo aver esaminato l'elenco dei tirocini, invia la richiesta di tirocinio all'applicazione Web, che, dopo aver controllato la sua identità, invia la domanda al Consiglio di corso di studi e nel contempo modifica le statistiche di richiesta di tirocinio; il Consiglio di corso di studi assegna il tirocinio allo studente e attribuisce allo stesso, una volta completato il tirocinio, una valutazione finale; se la valutazione risulta positiva, il tirocinio viene verbalizzato, altrimenti lo studente deve ripetere l'iter.

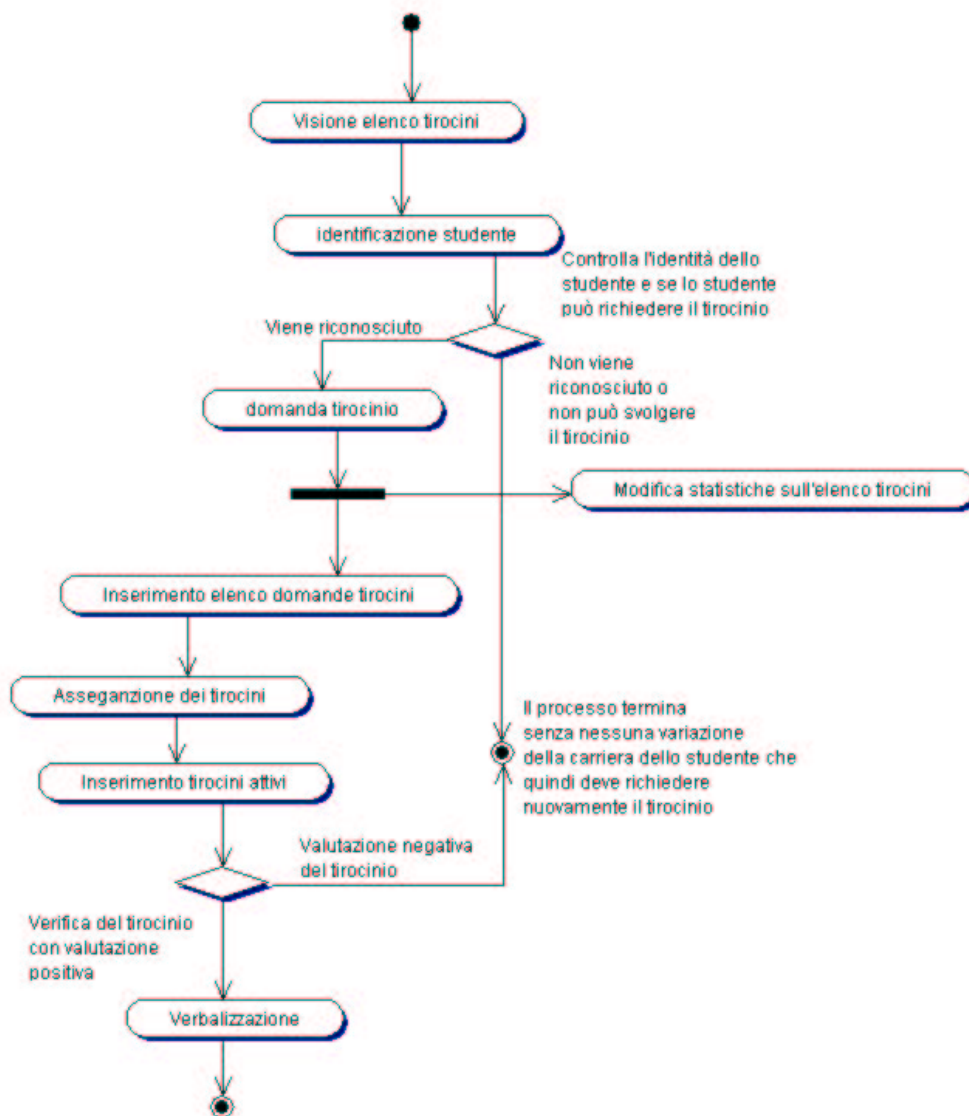


Figura 2.23: richiesta e assegnazione

2.3.3 Prova finale

In questo package viene definita una classe particolare chiamata Metadati. Questa classe elenca i dati fondamentali con cui può essere archiviata una prova finale. Alla pagina www.ndltd.org/standard/metadata/current.html del sito ndltd (Networked Digital Library of Theses and Dissertation) viene definito

uno standard per implementare i metadati. La classe del sistema di gestione delle prove finali ha preso spunto da questo standard, ma lo standard non è stato rispettato in tutti i suoi punti perché troppo particolareggiato per la gestione di tesi per un solo corso di studi.

Questo package gestisce l'archiviazione, la ricerca delle tesi di laurea in formato elettronico e l'elenco dei titoli delle prove finali.

Al suo interno sono definite le classi: Elenco_titoli, titoli_prova_finale, Controlla, Gestore_prove_finale, Archivio_prove_finale, Archivio_metadati, Metadati, Docente, Elenco_docenti.

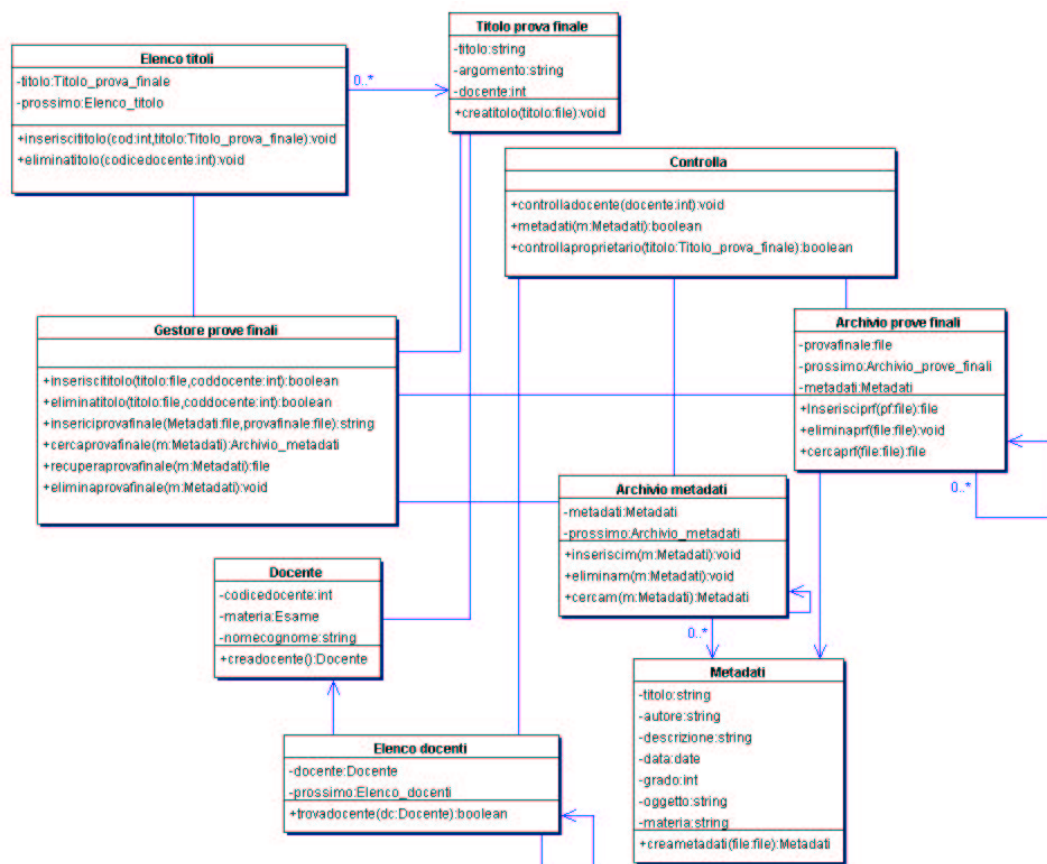


Figura 2.24: package prova finale

Descrizione dei compiti delle classi e dei loro metodi.

Gestore_prove_finale

Descrizione

È la classe fondamentale, gestisce tutte le operazioni del package prove finali.

Operazioni

inseriscititolo(): permette a un docente di inserire un titolo nell'elenco dopo che con il metodo controlladocente() è stata controllata l'esistenza del docente, viene inserito il titolo con il metodo inseriscititolo() della classe Elenco_titoli, che richiama creadocente().

eliminatitolo(): permette di eliminare un titolo dall'elenco titoli, per prima cosa controlla che il docente che ha inserito il titolo sia lo stesso che lo vuole eliminare e lo fa con il metodo controllaproprietario() della classe Controlla; poi invoca il metodo eliminatitolo() della classe Elenco_titoli.

cercaprovafinale(): permette di cercare una prova finale, indicando con indicazioni incomplete sui metadati, in output il metodo restituisce un oggetto della classe Archivio_metadati.

eliminaprovafinale(): permette di eliminare una prova finale dalla classe Archivio_prove_finali.

recuperaprovafinale(): permette di estrarre dall'archivio delle prove finali la tesi a cui corrispondono i metadati di input.

inserisciprovafinale(): permette di inserire la prova finale in archivio, richiamando il metodo metadati() della classe Controlla verifica che i metadati siano coerenti con il contenuto della tesi. Se il valore ritorna vero il metodo richiama inserisce la prova finale e i metadati in archivio, altrimenti torna un messaggio di errore.

Elenco_titoli

Descrizione

È una classe che definisce un elenco di oggetti della classe Titoli_prova_finale.

Operazioni

inseriscititolo(): inserisce un titolo nell'elenco.

eliminatitolo(): elimina un titolo dall'elenco.

Titolo_prova_finale

Descrizione

Definisce l'oggetto prova finale, quest'oggetto deve avere gli attributi: titolo, argomento, docente.

Operazioni

creatitolo(): permette di creare un oggetto della classe Titolo_prova_finale avendo come input un file.

Docente

Descrizione

Definisce un oggetto di tipo Docente, quest'oggetto deve avere gli attributi: codicedocente, materia, nome, cognome.

Operazioni

creadocente(): permette di creare un oggetto del tipo Docente dandogli in input un file.

Elenco _docenti

Descrizione

È una classe che definisce un elenco di oggetti della classe Docenti.

Operazioni

trovadocente(): permette di trovare un docente prendendo in input il codice-docente.

Metadati

Descrizione

Definisce un oggetto di tipo Metadati, quest'oggetto deve avere i seguenti attributi:

titolo, della prova finale

autore, chi ha scritto della prova finale

oggetto, le parti essenziali dell'argomento della prova finale

descrizione (una breve dichiarazione del lavoro svolto, la descrizione dell'abstract)

data, la data in cui è stata conclusa la prova finale

lingua, nella quale è stata scritta la prova finale

grado, se è una prova finale della laurea triennale, della laurea specialistica, del vecchio ordinamento.

materia: la materia di studio su cui è basata la prova finale

Operazioni

creametadati(): questo metodo permette di creare un oggetto del tipo Metadati prendendo in input un file.

Controlla

Descrizione

Definisce le operazioni di controllo per il package Prova finale.

Operazioni

controlladocente(): questo metodo permette di verificare che il docente sia nell'elenco dei docenti del corso di studi. Il metodo chiama il metodo trovadocente() della classe Elenco_docente. Se trova il docente il metodo restituisce il valore "true".

metadati(): questo metodo permette di controllare la coerenza tra la prova finale e i metadati.

controllaproprietario(): questo metodo permette di verificare che il codice docente (che viene preso in input dal metodo) sia uguale a quello dell'oggetto Titolo_prova_finale anch'esso preso in input.

Archivio_Metadati

Descrizione

È una classe che definisce un elenco di oggetti della classe Metadati.

Operazioni

inserisci(): permette di inserire un oggetto Metadati dall'archivio.

elimina(): permette di eliminare un oggetto della classe Metadati.

cerca(): permette di cercare dei metadati anche incompleti nell'archivio dei metadati.

Archivio_tesi

Descrizione

È una classe che definisce un elenco di Prove finali.

Operazioni

inserisci(): permette di inserire una prova finale nell'archivio.

elimina(): permette di eliminare una prova finale dall'archivio.

cerca(): permette di cercare una prova finale nell'archivio.

Diagrammi di sequenza

Consegna prova finale

Dopo che lo studente ha consegnato la prova finale anche in formato elettronico alla segreteria, un impiegato dovrà inserire i dati che gli ha fornito il laureando. L'impiegato, con il metodo `inserisciprovafinale()`, chiede al `Gestore_prove_finali` di inserire la prova finale nell'elenco; il `Gestore_prove_finali` con il metodo `metadati()`, verifica che la prova finale e i metadati siano coerenti; se torna vero, il `Gestore_prove_finali`, con il metodo `inseriscimetadati()`, inserisce i metadati e con il metodo `inserisciprovafinale()` inserisce nella classe `Archivio_prove_finali` la tesi.

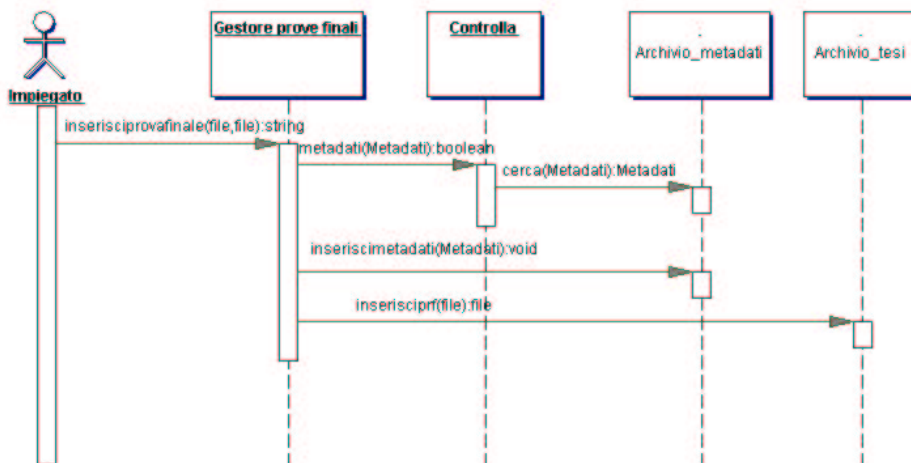


Figura 2.25: consegna prova finale

Inserimento titoli prova finale

Il docente, con il metodo `inseriscititolo()`, chiede al `Gestore_prove_finali` di inserire il titolo nell'`Elenco_titoli`; il `Gestore_prove_finali`, con il metodo `controlladocente()`, verifica che il docente sia nell'`Elenco_docenti`; se torna vero il valore, il `Gestore_prove_finali`, con il metodo `inseriscititolo()`, della classe `Elenco_titoli`, inserisce il titolo nell'elenco.

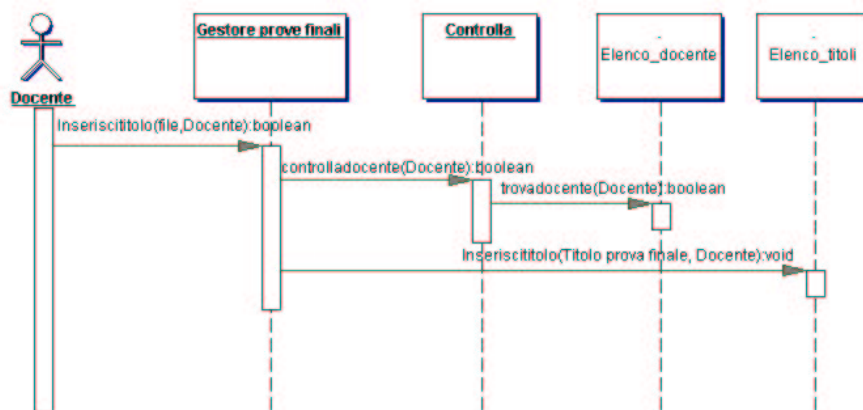


Figura 2.26: inserimento prova finale

Elimina titoli prova finale

Il docente, con il metodo `eliminatitolo()`, chiede al `Gestore_prove_finali` di eliminare il titolo dall'`Elenco_titoli`; il `Gestore_prove_finali`, con il metodo `controllaproprietario()`, verifica che il docente sia effettivamente il docente che ha inserito la prova finale nell'elenco titoli; se torna vero il valore, il `Gestore_prove_finali`, con il metodo `eliminatitolo()` della classe `Elenco_titoli`, elimina il titolo dall'elenco.

Diagramma delle attività

Scelta prova finale

Lo studente si collega alla pagina Web, vede quale potrebbe essere il titolo della prova finale che più gli piace e poi va dal docente a discuterne. Se lo studente sceglie di fare la prova finale il docente provvederà all'eliminazione del titolo dall'elenco delle prove finali.

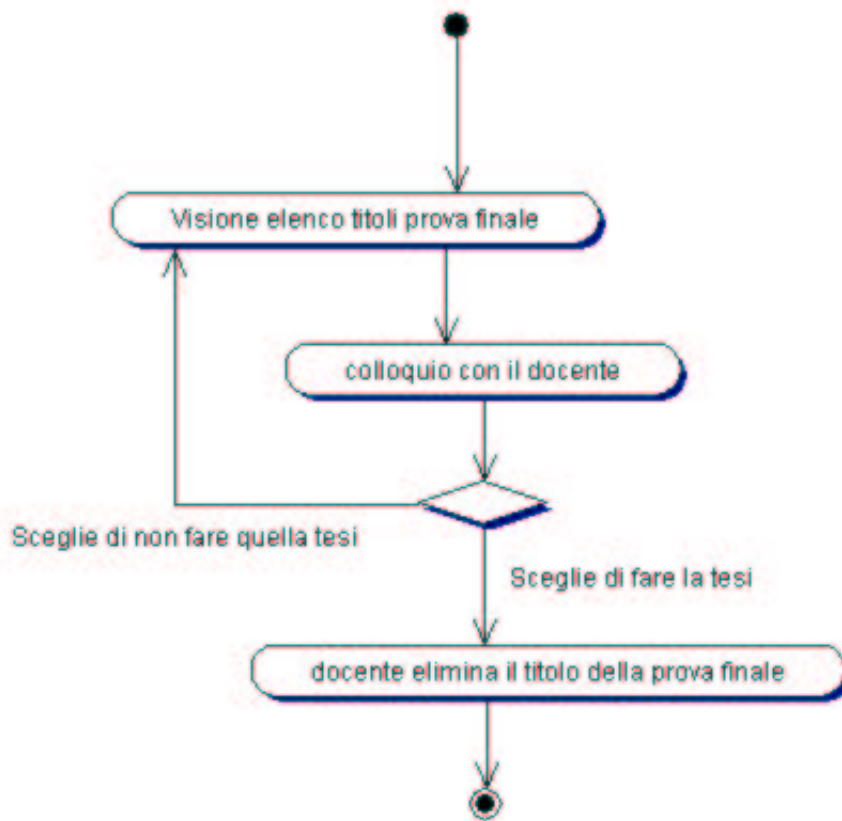


Figura 2.27: scelta prova finale

Inserimento di un titolo di prova finale

Il docente inserisce il titolo della prova finale, viene aggiornato l'elenco dei titoli; a questo punto il docente può compiere un'altra operazione di inserimento o di eliminazione, altrimenti può uscire dal sistema

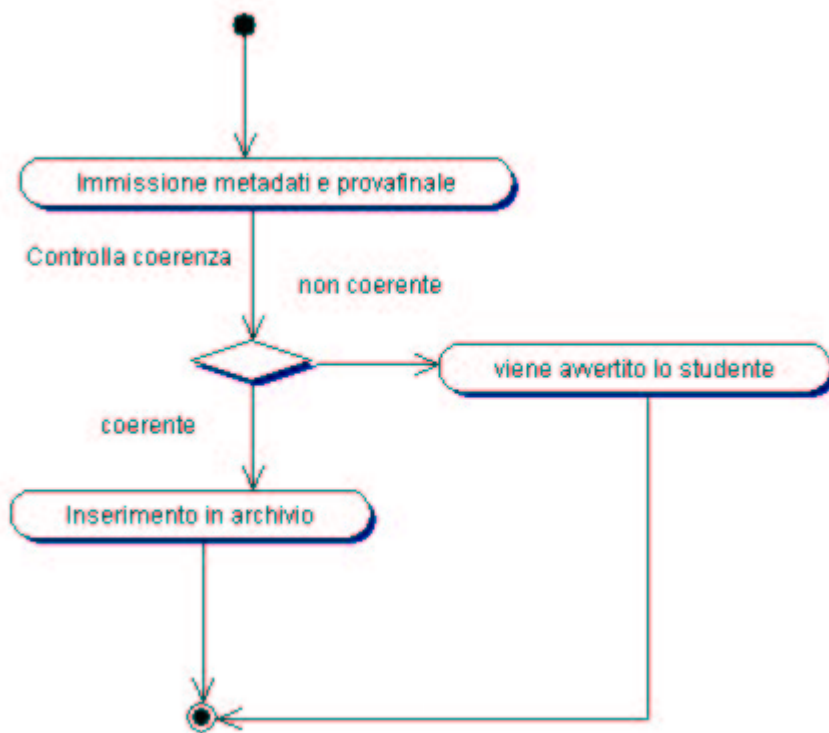


Figura 2.28: inserimento titolo prova finale

Inserimento prova finale

L'impiegato, dopo aver ricevuto la prova finale in formato elettronico dallo studente, invia la richiesta di inserimento in archivio della prova finale all'applicazione Web, che controlla che i metadati e la prova finale siano coerenti nei contenuti; se risultano coerenti vengono inseriti nell'archivio altrimenti viene avvertito lo studente.

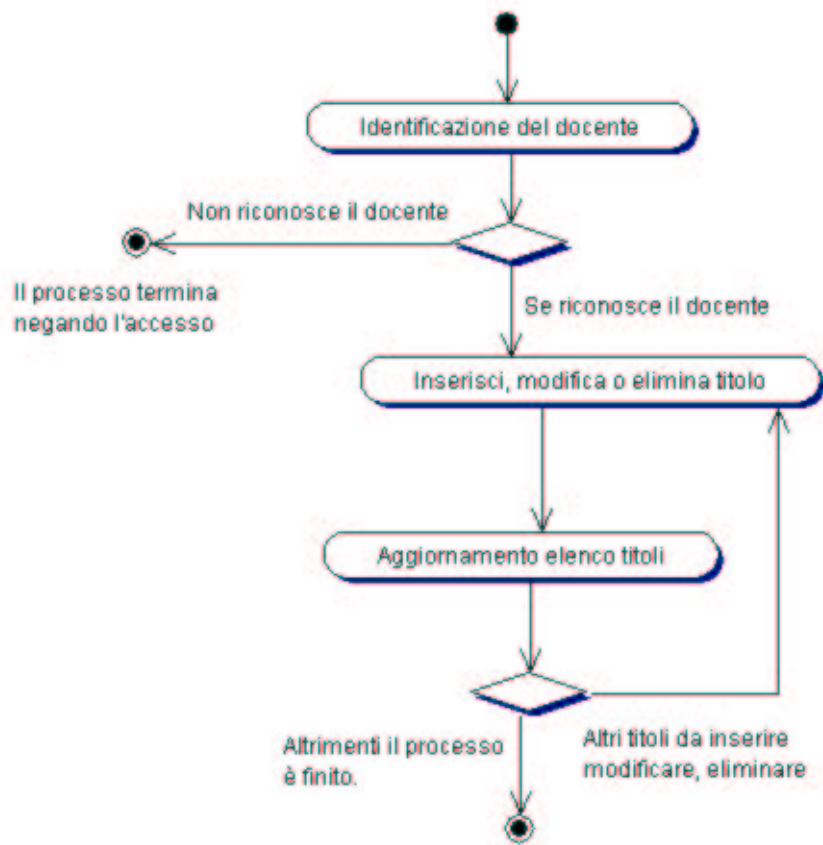


Figura 2.29: inserimento prova finale

Appendice A

Appendice

Estensione di UML per applicazioni Web descritta da Jim Conallen in [JC 02]

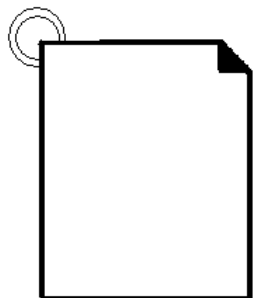
Stereotipi

Server Page

Classe del metamodello Classe

Descrizione Una pagina server rappresenta una pagina Web che ha script eseguiti dal server. Questi script interagiscono con le risorse del server quali basi di dati, logica di business e sistemi esterni. Le operazioni dell'oggetto rappresentano le funzioni nello script e i suoi attributi rappresentano le variabili che sono visibili nella pagina (accessibili da tutte le funzioni nella pagina).

Icona



Vincoli Le pagine server possono avere relazioni solo con oggetti sul server.

Valori etichetta Motore di script: il linguaggio o il motore che dovrebbe essere usato per eseguire o interpretare questa pagina (JavaScript, VBScript, PERL e così via).

Client Page

Classe del metamodello Classe

Descrizione Un'istanza di una pagina client è una pagina Web formattata in HTML ed è un misto di dati, presentazione e persino logica. Le pagine client sono visualizzate dai browser client e potrebbero contenere script che sono interpretati dal browser. Le funzioni della pagina client corrispondono a funzioni nei tag di script nella pagina. Gli attributi della pagina client corrispondono a variabili dichiarate nel tag di script della pagina accessibili da qualsiasi funzione nella pagina (visibilità della pagina). Le pagine client possono avere associazioni con altre pagine client o server.

Icona

Vincoli Nessuno

Valori etichetta

- TitleTag: il titolo della pagina come visualizzato dal browser.
- BaseTag: l'URL di base per definire le URL relative.

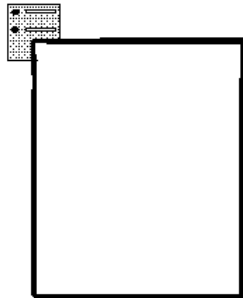
- BodyTag: l'insieme di attributi per il tag <body>, che imposta gli attributi per lo sfondo (background) e il testo standard.

Form

Classe del metamodello Classe

Descrizione Una classe stereotipata come una <<form>> è un insieme di campi di input che sono parte di una pagina client. Una classe modulo (form) corrisponde direttamente al tag <<form>> di HTML. Gli attributi di questa classe rappresentano i campi di input per un modulo HTML (input box, text area, radio button, check box e hidden field) Una <<form>> non ha operazioni, poiché queste non possono essere incapsulate in un modulo. Qualunque operazione che interagisce con il modulo è proprietà della pagina che contiene il modulo.

Icona



Vincoli Nessuno

Valori etichetta Il metodo - GET o POST - usato per sottomettere i dati all'URL di action.

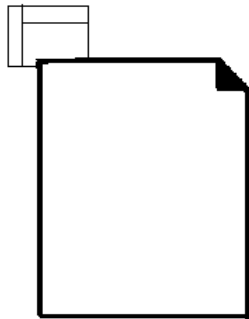
Frameset

Classe del metamodello Classe

Descrizione Un frameset è un contenitore di più pagine Web. L'area di visualizzazione rettangolare è divisa in frame rettangolari più piccoli. Ogni frame potrebbe essere associato a un singolo <<target>> con nome, sebbene

questo non sia necessario. Il contenuto di un frame potrebbe essere una pagina Web o un altro frameset. Una classe stereotipata come frameset corrisponde direttamente a una pagina Web frameset e al tag frame di HTML. Poiché un frameset è una <<client page>>, anch'esso può avere operazioni e attributi, ma questi sono attivati solo dai browser che non rappresentano i frame.

Icona



Vincoli Nessuno

Valori etichetta

- Rows: il valore dell'attributo rows (righe) del tag <frameset> di HTML. Questa stringa specifica le dimensioni delle righe, separate da virgole.
- Cols: il valore dell'attributo cols (colonne) del tag <frameset> di HTML. Questa stringa specifica le dimensioni delle colonne, separate da virgole.

Target

Classe del metamodello Classe

Descrizione Un target è un compartimento con nome in una finestra del browser in cui possono essere visualizzate le pagine Web. Il nome della classe stereotipata è il nome del target. Tipicamente, un target è un frame in una finestra definita da un frameset; comunque, un target potrebbe essere un'istanza o una finestra del browser completamente nuova. Le associazioni <<targeted link>> specificano i target come il punto in cui dovrebbe essere visualizzata una nuova pagina Web.

Icona

Vincoli Il nome di un target deve essere univoco per ogni client del sistema. Perciò, solo un'istanza di un target può esistere sul medesimo client.

Valori etichetta Nessuno

JavaScript Object

Classe del metamodello Classe

Descrizione Su un browser JavaScript, è possibile simulare oggetti definiti dall'utente con funzioni JavaScript. Istanze di <<JavaScript Object>> esistono solo nel contesto di pagine client.

Icona Nessuna

Vincoli Nessuno

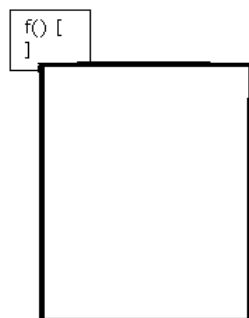
Valori etichetta Nessuno

ClientScript Object

Classe del metamodello Classe

Descrizione Un ClientScript Object è un insieme separato di script lato client che esistono in un file separato e sono inclusi con una richiesta separata del browser client. Questi oggetti sono spesso insiemi di funzioni usate comunemente nell'applicazione o dall'azienda. Alcuni browser sono in grado di memorizzare nella cache tali file, riducendo i tempi di scaricamento (download) complessivi di un'applicazione.

Icona



Vincoli Nessuno

Valori etichetta Nessuno

Link

Classe del metamodello Associazione

Descrizione Un link è un puntatore da una pagina client a un'altra <<Page>> (pagina). In un diagramma delle classi, un link è un'associazione tra una <<client page>> e un'altra <<client page>> o una <<server page>>. Un link corrisponde direttamente al tag ancora di HTML.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Parameters (parametri): un elenco di nomi di parametri che dovrebbero essere passati con la richiesta per la pagina a cui punta il link.

Targeted Link

Classe del metamodello Associazione

Descrizione Simile a un'associazione <<link>>, un <<targeted link>> è un link la cui pagina associata è visualizzata in un altro target. Quest'associazione corrisponde direttamente al tag ancora di HTML, con il target specificato dall'attributo target del tag.

Icona Nessuna

Vincoli Nessuno

Valori etichetta

- Parameters: un elenco di nomi di parametri che dovrebbero essere passati con la richiesta per la pagina a cui punta il link.
- Target Name (nome del target): il nome del <<target>> in cui dovrebbe essere visualizzata la pagina alla quale questo link punta.

Frame Content

Classe del metamodello Associazione

Descrizione Un'associazione frame content è un'associazione di aggregazione che esprime il contenimento di una pagina o target da parte di un frame. Un'associazione frame content può anche puntare a un altro frameset, indicando frame annidati.

Icona Nessuna

Vincoli Nessuno

Valori etichetta

- Row: un intero che indica la particolare riga del frame, nel frameset, a cui appartiene la pagina o il target associato.
- Col: un intero che indica la particolare colonna del frame, nel frameset, a cui appartiene la pagina o il target associato.

Submit

Classe del metamodello Associazione

Descrizione Un'associazione <<submit>> è sempre tra un <<form>> e una <<server page>>. I moduli inviano i valori dei loro campi al server attraverso <<server pages>>. Il server Web elabora la <<server page>> che accetta e utilizza le informazioni nel modulo sottomesso. Questa relazione indica quale pagina (o pagine) può elaborare il modulo e di quali moduli una <<server page>> è a conoscenza.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Parameters: un elenco di nomi di parametri che dovrebbero essere passati con la richiesta per la pagina cui punta il link.

Build

Classe del metamodello Associazione

Descrizione La relazione <<build>> è una relazione speciale che colma il divario tra pagine client e server. Le pagine server esistono solo sul server. Sono usate per realizzare le pagine client. L'associazione <<build>> identifica quale pagina server è responsabile della creazione di una pagina client.

Questa è una relazione monodirezionale, poiché la pagina client non ha alcuna conoscenza di come ha avuto origine. Una pagina server può realizzare più pagine client, ma una pagina client può essere realizzata da una sola pagina server.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Nessuno

Redirect

Classe del metamodello Associazione

Descrizione Una relazione <<redirect>>, un'associazione unidirezionale con un'altra pagina Web, può essere diretta sia da sia verso pagine client e server.

Se la relazione prende origine da una <<server page>>, l'elaborazione della richiesta di pagina potrebbe continuare con l'altra pagina. Questo non indica che la pagina destinazione partecipa sempre alla realizzazione di una pagina client, ma potrebbe accadere. Questa particolare relazione non è completamente strutturale, poiché l'invocazione di un'operazione di ridirezione deve essere programmata nel codice della pagina di origine. Se la relazione prende origine da una <<client page>>, questo indica che la pagina di destinazione sarà richiesta automaticamente dal browser, senza l'input dell'utente. È possibile impostare un ritardo, in secondi, prima che la seconda pagina venga richiesta. Questo utilizzo della ridirezione corrisponde al tag <<META>> e al valore HTTP - EQUIV di Refresh.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Delay: la quantità di tempo che una pagina client dovrebbe attendere prima di rivolgersi alla pagina riferita. Questo valore corrisponde all'attributo Content del tag <<META>>.

IIOP

Classe del metamodello Associazione

Descrizione IIOP (Internet Inter-ORB Protocol) è un tipo speciale di relazione tra oggetti sul client e oggetti sul server. IIOP rappresenta un meccanismo diverso da HTTP per comunicazioni client/server. Tipicamente, questa relazione è tra JavaBeans sul client e Enterprise JavaBeans sul server.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Nessuno

RMI

Classe del metamodello Associazione

Descrizione RMI (Remote Method Invocation) è un meccanismo per applet Java e JavaBeans per inviare messaggi ad altri JavaBeans su macchine differenti. Tipicamente, questa relazione è tra JavaBeans o applet sul client e Enterprise JavaBeans sul server.

Icona Nessuna

Vincoli Nessuno

Valori etichetta Nessuno

Input Element

Classe del metamodello Attributo

Descrizione Un input element, un attributo di un oggetto <<Form>> corrisponde direttamente al tag <input> in un modulo di HTML. Questo attributo è usato per inserire una parola singola o una linea di testo.

I valori etichetta associati a questo attributo stereotipato corrispondono agli attributi del tag <input>. Per completare i valori richiesti per il tag HTML, il nome dell'attributo è usato come nome del tag <input> e il valore iniziale dell'attributo è il valore del tag.

Icona Nessuna

Vincoli Nessuno

Valori etichetta

- **Type:** il tipo di controllo di input da usare: text, number, password, checkbox, radio, submit, oreset.

- *Size*: specifica la dimensione, in caratteri, dell'area da allocare sullo schermo.
- *Maxlength*: il numero massimo di caratteri che l'utente può inserire.

Select Element

Classe del metamodello Attributo

Descrizione Un controllo di input usato nei moduli. Questo controllo permette all'utente di selezionare uno o più elementi da una lista. La maggior parte dei browser visualizzano questo controllo come una combo o list box.

Icona Nessuna

Vincoli Nessuno

Valori etichetta

- *Size*: specifica quanti elementi dovrebbero essere visualizzati contemporaneamente.
- *Multiple*: un valore booleano che indica se è possibile selezionare più elementi della lista.

Text Area Element

Classe del metamodello Attributo

Descrizione Un controllo di input, usato nei moduli, che permette l'inserimento di più linee di testo.

Icona Nessuna

Vincoli Nessuno

Valori etichetta

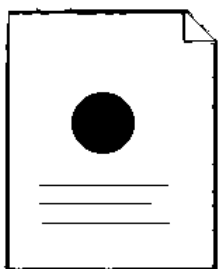
- *Rows*: il numero di linee di testo visibili.
- *Cols*: l'ampiezza visibile del controllo usando un carattere di dimensioni medie.

Web Page

Classe del metamodello Componente

Descrizione Un componente pagina è una pagina Web. Può essere richiesta da un browser per nome. Un componente pagina potrebbe contenere o meno script client o server. Tipicamente, i componenti pagina sono file di testo accessibili dal server Web, ma possono anche essere moduli compilati che sono caricati e invocati dal server Web. Alla fine, quando il server Web accede alla pagina, questa produce un documento formattato in HTML che è inviato in risposta a una richiesta del browser.

Icona



Vincoli Nessuno

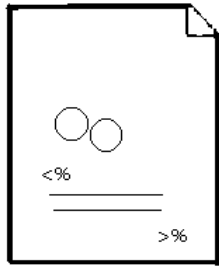
Valori etichetta Path: il percorso richiesto per specificare la pagina Web sul server Web. Questo valore dovrebbe essere relativo alla directory radice dell'applicazione Web (del sito).

ASP Page

Classe del metamodello Componente

Descrizione Le pagine Web che implementano codice ASP lato server. Questo stereotipo è applicabile solo in ambienti basati su Microsoft Active Server Pages.

Icona



Vincoli Nessuno

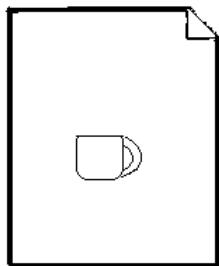
Valori etichetta Stessi valori etichetta come per la pagina Web

JSP Page

Classe del metamodello Componente

Descrizione Pagine Web che implementano codice JSP lato server. Questo stereotipo è applicabile solo in ambienti di sviluppo di applicazioni Web che utilizzano Java Server Pages.

Icona



Vincoli Nessuno

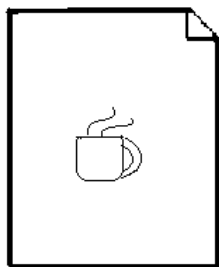
Valori etichetta Stessi valori etichetta come per la pagina Web

Servlet

Classe del metamodello Componente

Descrizione Un componente Java servlet. Questo stereotipo è applicabile solo in ambienti di sviluppo di applicazioni Web che gestiscono servlet Sun.

Icona



Vincoli Nessuno

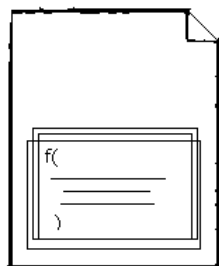
Valori etichetta Stessi valori etichetta come per la pagina Web

Script Library

Classe del metamodello Componente

Descrizione Il componente che fornisce una libreria di subroutine o funzioni che possono essere incluse in altre pagine Web.

Icona



Vincoli Nessuno

Valori etichetta Stessi valori etichetta come per la pagina Web

Bibliografia

- [GMb 01] The Essence of Web Engineering”, *IEEE MultiMedia*, 22, 25, April - June 2001
- [GMa 01] Athula Ginige, San Murugesn, “Web Engineering: An Introduction”, *IEEE MultiMedia*, 14, 18, January - March 2001
- [TAP 98] Thomas A. Powell, *Web Site Engineering*, Pearson Education, 1998
- [KST 00] Christoph Kuhnke, Josef Schneeberger, Andreas Turk, “A Schema-Based Approach to Web Engineering”, *Proceedings of the 33rd Hawaii International Conference on System Sciences - 2000*
- [BGP 01] Luciano Baresi, Franca Garzotto, end Paolo Paolini, “Extending UML for Modeling Web Applications”, *Proceedings of the 34th Hawaii International Conference on System Sciences - 2001*
- [U 02] Loran Uden, “Design Process for Web Application”, *IEEE MultiMedia*, 47, 55 October-December 2002
- [AN 02] Jim Arlow, Ila Neustadt, *UML and the unified process practical object-oriented analysis and design*, Pearson Education Limited, 2002
- [MH 98] Laura G. Militello, Robert J. B. Hutton, "Applied cognitive task analysis: a pracioner’s toolkit for understanding cognitive demands", *Ergonomics*, 1998 vol. 41 so. 11, 1618 - 1641
- [JC 02] Jim Conallen, *Applicazioni web con UML*, Addison Wesley, 2002

- [S 92] B. Shneiderman, *Designing the user interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Boston, 1992
- [FS 00] M. Fowler, K. Scott, *UML Distilled*, Addison-Wesley, 2000
- [RP 3] R. Pressman, *Principi di Ingegneria del Software*, 3a edizione, McGrawHill
- [SRB 96] D. Schwabe, G. Rossi, and S. Barbarossa, "Systematic Hyperbase Design whit OOHDM", *Proc. ACM Int'l Conf Hypertext* (Hypertext 96), ACM Press, New York, 1996, pag. 45, 46
- [SR 98] D. Schwabe, G. Rossi, "An Object Oriented Approach to Web-Based Application Design," *TAPoS: Theory and Practice of Object System*, vol.4 no. 4, Oct. 1998, pag. 207- 225
- [JB 00] Joonhee Yoo and Micheal Bieber, "Finding Relationship-Based Analysis", *Proc. Hypertex 2000*, ACM Press, New York, 2000, pp. 181, 190
- [W1] www.menesbatto.com
- [W2] www.ndltd.org
- [W3] www.i-scences.net
- [W4] www.unipr.it/studenti/laureati/tesi
- [W5] <http://master2.cci.unibs.it/cgi-bin/tesi/ricerca.pl>
- [W6] www.scwww.scform.unibo.it