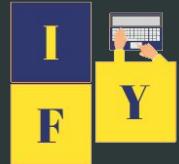


IFY - Internship For You - Maintrance Project



Corso: Ingegneria, Gestione ed Evoluzione del Software

Anno Accademico: 2020/2021

Docente: Professor Andrea De Lucia

Team di Manutenzione:

Francesco Abate - 0522500993

Ferrara Carmine - 0522500990

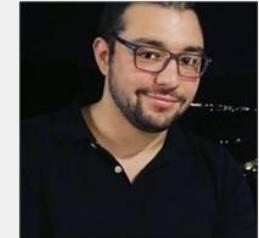
Gianluca Spinelli - 0522500981



Francesco Abate

Email: f.abate20@studenti.unisa.it

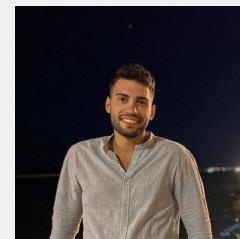
GitHub: frekkanzer2



Carmine Ferrara

Email: c.ferrara49@studenti.unisa.it

GitHub: CFerrara98



Gianluca Spinelli

Email: g.spinelli18@studenti.unisa.it

GitHub: GianlucaSpinelli



INTRODUZIONE AD IFY

IFY è una piattaforma di supporto agli studenti interessati ad un'offerta di tirocinio esterno mirata allo svolgimento di un progetto formativo con le aziende

IFY offre, quindi, supporto all'ufficio tirocini del dipartimento di informatica, automatizzando il processo di convenzionamento aziendale e di richiesta tirocinio da parte dello studente

Il sistema sviluppato permette di far convenzionare nuove aziende con il dipartimento, di pubblicare progetti formativi che saranno visualizzabili dagli studenti e di inviare richieste di tirocinio per un particolare progetto formativo

PROBLEMATICHE DI IFY

La gestione delle richieste di tirocinio garantita dal prodotto a seguito del suo sviluppo prevede una doppia approvazione delle richieste di tirocinio:

- Approvazione aziendale da parte del referente
- Approvazione accademica da parte dell'ufficio tirocini

La singola approvazione e supervisione accademica effettuata dall'ufficio non corrisponde, però, al processo di richiesta e approvazione del tirocinio realmente applicata dal dipartimento

L'iter di gestione di un tirocinio esterno prevede che alla presentazione di una richiesta di tirocinio si faccia riferimento anche ad un designato tutor accademico.

Lo scopo del progetto di manutenzione sarà quello di adattare l'iter attuale del sistema al reale iter sostenuto dal dipartimento



Maintenance Report

INTRODUZIONE ALLE CR

Una prima change request vuole che si aggiunga un nuovo utente funzionale, quale è il Tutor accademico

Il Tutor accademico dovrà occuparsi della supervisione e dell'approvazione delle richieste di tirocinio

- Gestione delle richieste di tirocinio presentate dagli studenti
- Accesso alla propria area personale dedicata con visualizzazione di tutte le richieste di tirocinio

Una seconda change request vuole che il Tutor accademico possa terminare l'esperienza di tirocinio degli studenti prima che scada sulla piattaforma

Il Tutor accademico potrà terminare l'esperienza di tirocinio evitando di dover attendere la chiusura automatica di quest'ultima da parte del sistema

- Chiusura di un'esperienza di tirocinio dalla propria area riservata
- Visualizzazione delle esperienze di tirocinio contrassegnate in base alla terminazione con successo o in base alla scadenza

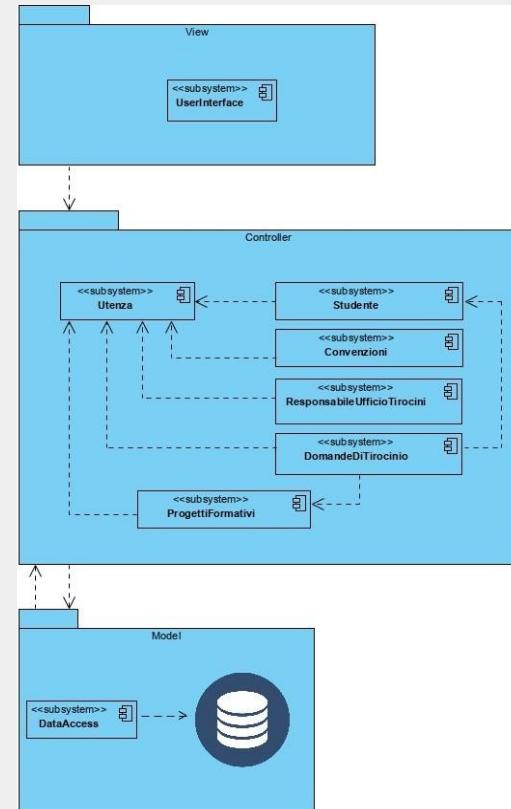
PUNTI CHIAVE DELL'ARCHITETTURA ATTUALE

L'architettura attuale è basata sul modello MVC e include i seguenti packages:

- View:
 - UserInterface
- Model:
 - DataAccess
- Controller:
 - Utenza
 - Studente
 - Convenzioni
 - ResponsabileUfficioTirocini
 - DomandeDiTirocinio
 - ProgettiFormativi

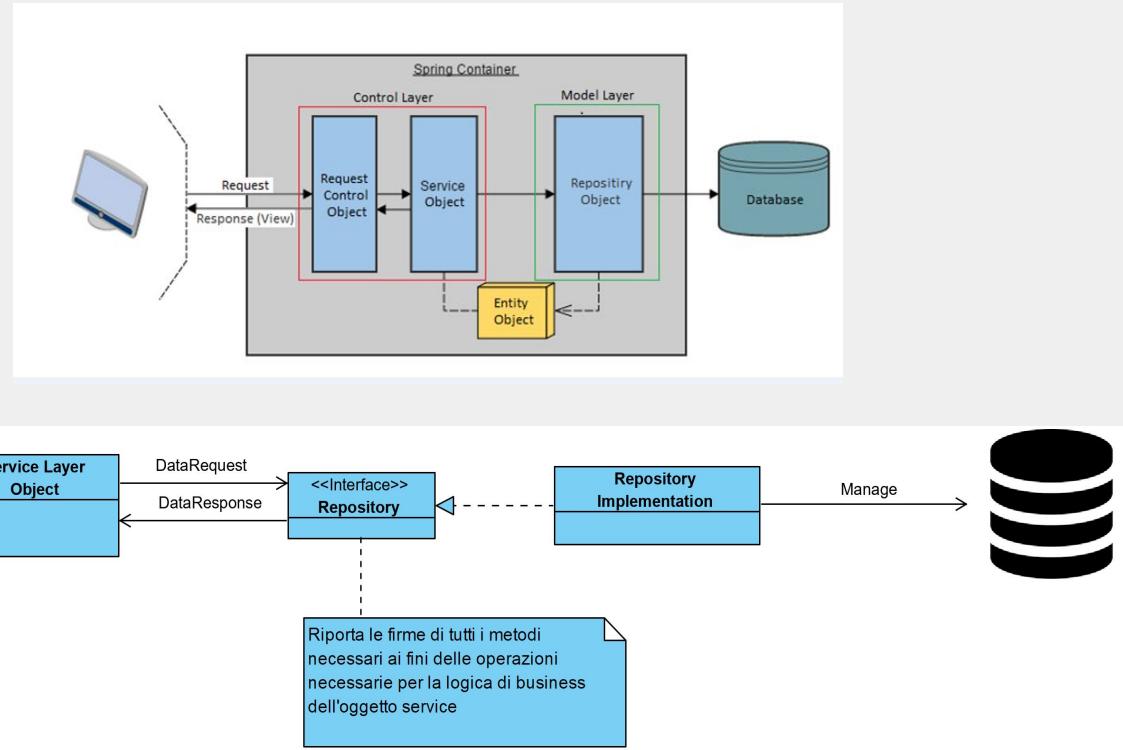
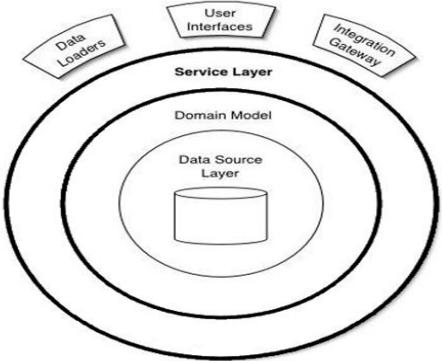
Le richieste di modifica si dovranno adattare all'architettura già esistente andando a modificare nello specifico i packages di controllo, utenza, domande di tirocinio e introducendo un nuovo package relativo alle funzionalità del docente Tutor.

Conseguentemente alle modifiche di controllo, andranno modificati anche i packages di user interface e di persistenza.





DESIGN PATTERN DI RILIEVO





Change Request 1

DIAGRAMMA DELLE ATTIVITÀ

Prima e dopo

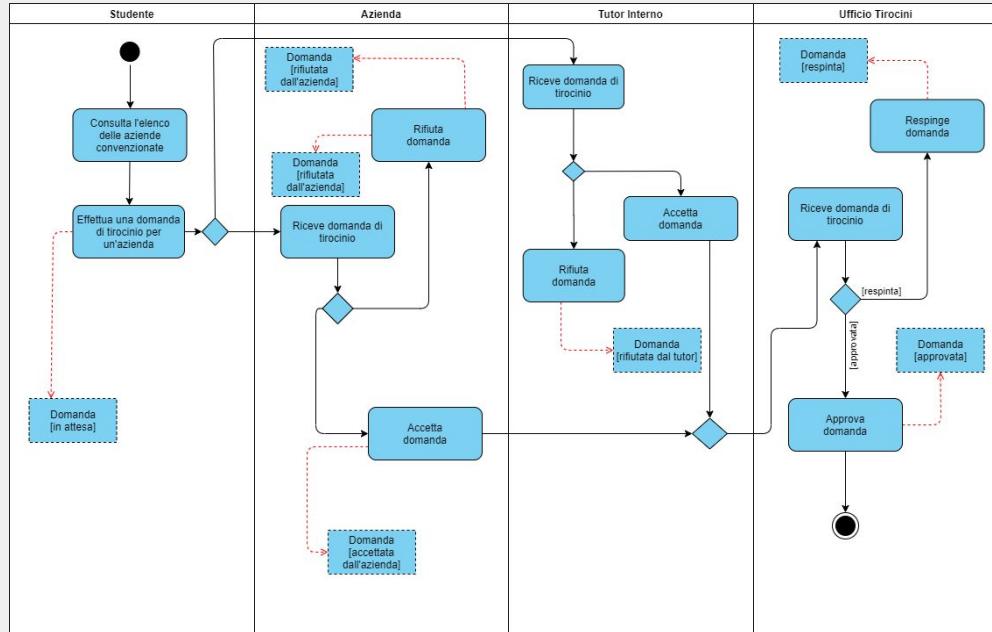
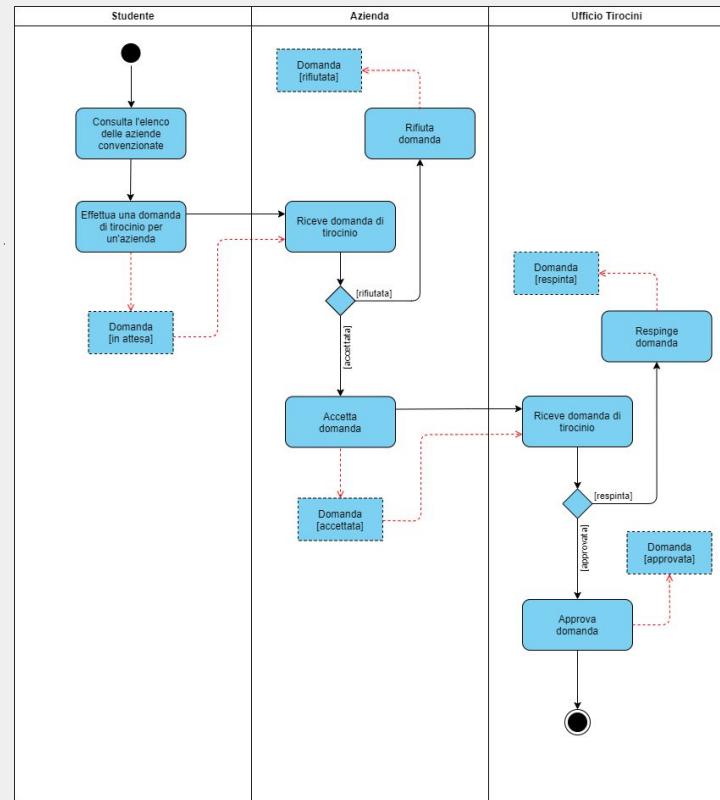
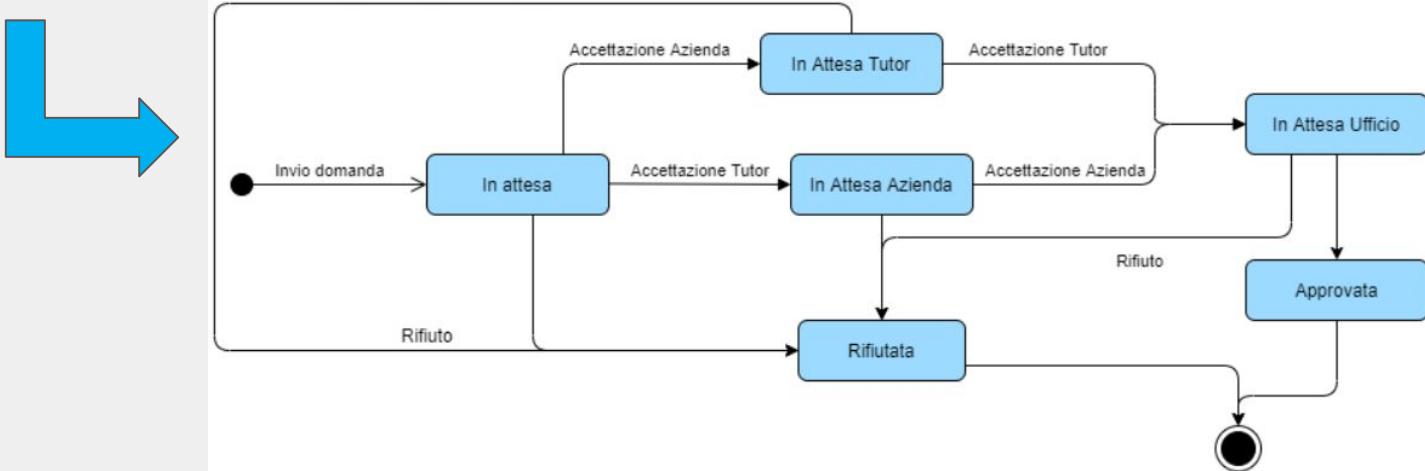
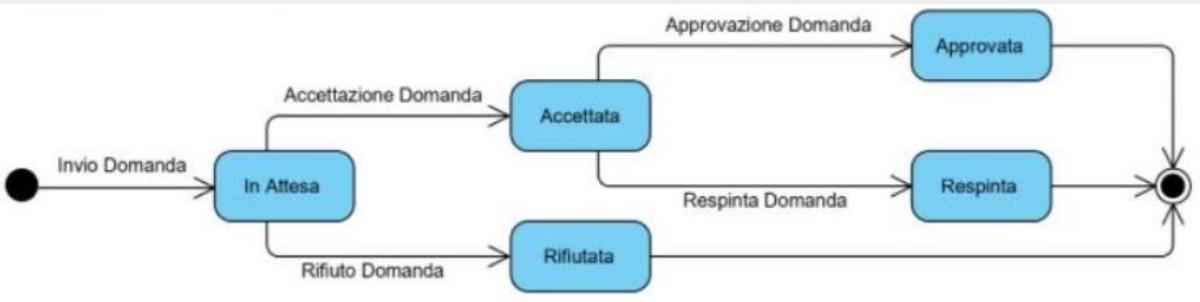


DIAGRAMMA DEGLI STATI

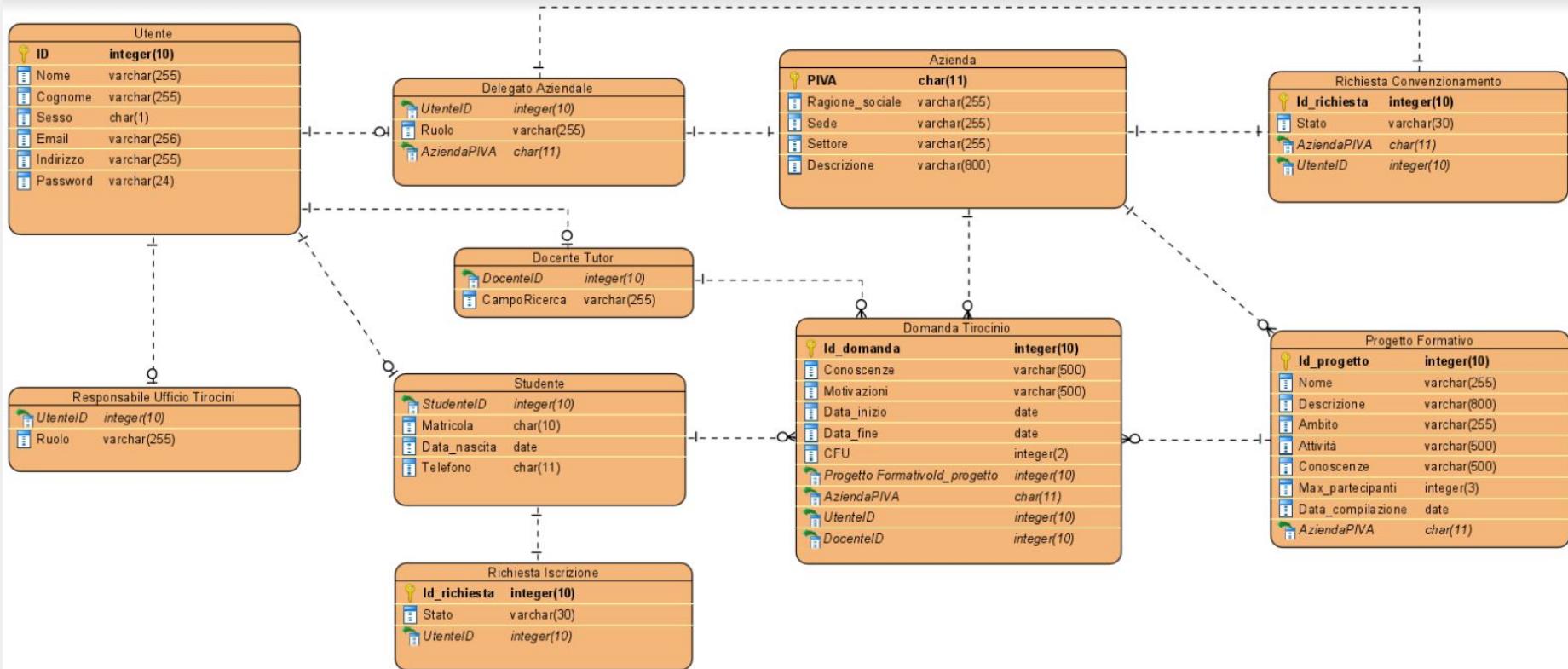
Prima e dopo



SCHEMA DI PERSISTENZA (PRIMA)



SCHEMA DI PERSISTENZA (DOPO)





IMPACT ANALYSIS



Le modifiche per la realizzazione della prima change request sono precedute da un'analisi di impatto sulle componenti già esistenti nel sistema.

In particolare, lo starting impact set ed il candidate impact set (SIS e CIS) sono stati formulati secondo i seguenti principi:

- Introduzione del nuovo utente funzionale Docente Tutor;
- Introduzione dei metodi di servizio per il Docente Tutor;
- Modifiche all'entità Domanda di Tirocinio in relazione ai nuovi stati individuati comportati dall'introduzione del nuovo utente;
- Aggiunta e modifica di metodi di servizio e di controllo per il flusso della domanda di tirocinio in relazione alle relazioni con il nuovo utente funzionale;
- Specializzazione del metodo di accesso per permettere l'autenticazione al Docente Tutor.

Le credenziali di accesso al sistema per il Docente Tutor vengono fornite dal dipartimento, quindi non previa registrazione.

IMPACT ANALYSIS

CANDIDATE IMPACT SET - 55 componenti impattate

ACTUAL IMPACT SET - 58 componenti impattate

FALSE POSITIVE IMPACT SET - 0 componenti impattate

DISCOVERED IMPACT SET - 3 componenti impattate

Recall: $|CIS \cap AIS| \div |AIS| = 55/58 = 0.95$;

Precision: $|CIS \cap AIS| \div |CIS| = 55/55 = 1$;

Inclusiveness = 0

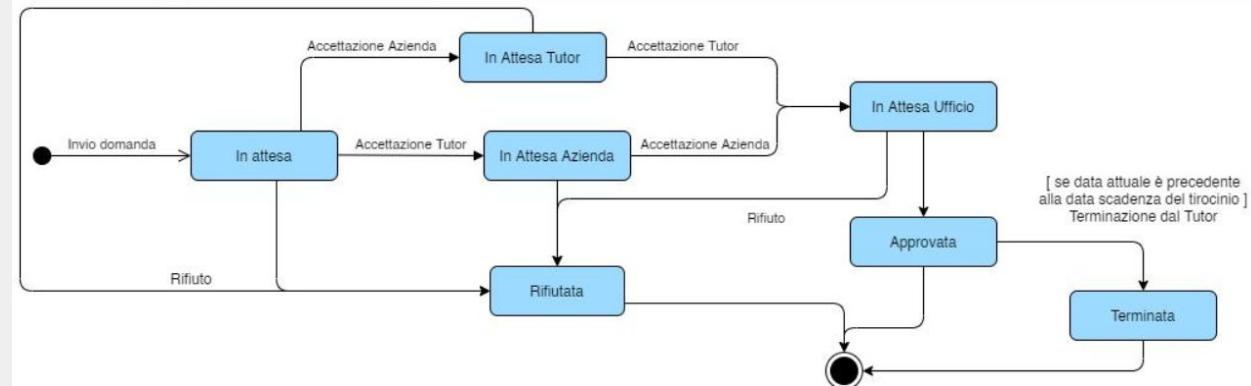
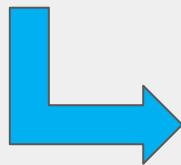
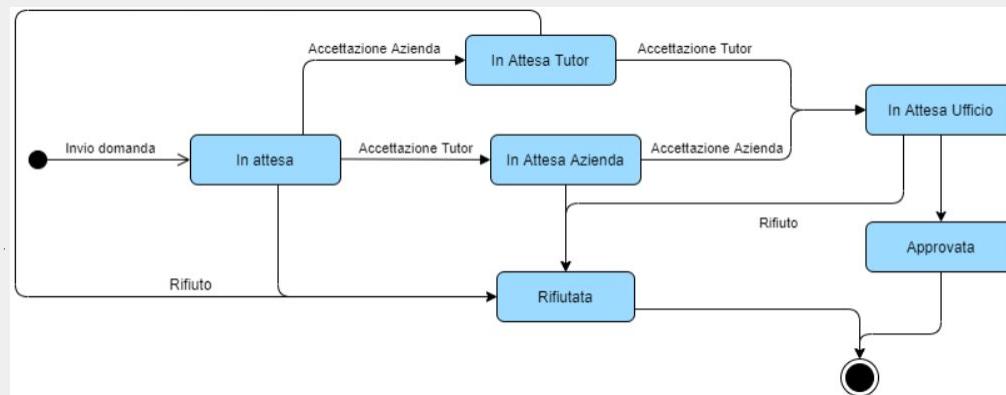




Change Request 2

DIAGRAMMA DEGLI STATI

Prima e dopo



IMPACT ANALYSIS

Le modifiche per la realizzazione della seconda change request sono precedute da un'analisi di impatto sulle componenti già esistenti nel sistema.

In particolare, lo starting impact set ed il candidate impact set (SIS e CIS) sono stati formulati secondo i seguenti principi:

- Introduzione di un nuovo stato per differenziare la scadenza del tirocinio dalla chiusura di quest'ultimo da parte del Tutor con successo;
- Specializzazione del Mail Sender per inviare un messaggio di conferma nel caso il tirocinio venga chiuso con successo;
- Aggiunta di metodi di servizio e controllo per la chiusura del tirocinio.

IMPACT ANALYSIS

CANDIDATE IMPACT SET - 6 componenti impattate

ACTUAL IMPACT SET - 7 componenti impattate

FALSE POSITIVE IMPACT SET - 0 componenti impattate

DISCOVERED IMPACT SET - 1 componente impattata

Recall: $|CIS \cap AIS| \div |AIS| = 6/7 = 0.86$;

Precision: $|CIS \cap AIS| \div |CIS| = 6/6 = 1$;

Inclusiveness = 0



Implementazione e Testing



COMPONENTI UTILIZZATE

Le componenti usate per l'implementazione sono:

- Back-end: Spring, Java, Maven, MySQL
- Front-end: HTML5, jQuery, jstl, JavaScript, CSS3, Bootstrap4



JavaScript



VERSIONING E CONFIGURATION MANAGEMENT



GITHUB

Il codice è stato condiviso con i vari team members tramite l'uso di GitHub

main 1 branch 0 tags Go to file Add file Code

GianlucaSpinelli Update README.md 97b978d 7 minutes ago 42 commits

.mvn/wrapper load starting project materials 4 months ago

src Update system-test-inviandomandatiocinio.side last month

.gitignore load starting project materials 4 months ago

README.md Update README.md 7 minutes ago

mvnw load starting project materials 4 months ago

mvnw.cmd load starting project materials 4 months ago

pom.xml load starting project materials 4 months ago

README.md

ify-iges

Repository Progettuale Esame IGES 2020/2021 - Abate Francesco / Ferrara Carmine / Spinelli Gianluca Il progetto consiste nell'effettuare modifiche evolutive e correttive del progetto IFY tramite opportuno processo di manutenzione, che dal repository originale rilasciato dal precedente team di sviluppo è così descritto:

IFY

Applicazione web per la gestione dei tirocini formativi sviluppata per il Dipartimento di Informatica dell'Università degli Studi di Salerno per il corso di Ingegneria del Software 2019/2020

La documentazione iniziale del progetto e quella prodotta ai fini del progetto di manutenzione è disponibile su google drive al link: <https://drive.google.com/drive/u/1/folders/0AEltgEy1JD8FUk9PVA>.

About Repository Progettuale Esame IGES 2020/2021 - Abate Francesco / Ferrara Carmine / Spinelli Gianluca Readme

Releases No releases published Create a new release

Packages No packages published Publish your first package

Contributors 3 frekanzer2 Francesco Abate CFerrara98 Carmine Ferrara GianlucaSpinelli

Languages Java 62.8% CSS 34.3% JavaScript 2.9%



TESTING DI UNITÀ E DI INTEGRAZIONE

- Junit
- Mockito
- Jacoco



TESTING DI SISTEMA

- Selenium Ide





TESTING DI UNITÀ

Approcci utilizzati:

- White-box testing con tecnica Branch Coverage

Componenti Testate:

- DomandaDiTirocinioService
- DomandaDiTirocinioRepository
- DomandaDiTirocinio
- DocenteTutorService
- DocenteTutorRepository
- DocenteTutor

```
ify-iges D:\unisa\2.Magistrale\IGES\Progetto\ify-iges
> .idea
> .mvn
> src
  > main
    > java
      > it.unisa.di.is.gc1.ify
        > convenzioni
        > DocenteTutor
          > DocenteTutor
          > DocenteTutorRepository
          > DocenteTutorService
      > domandaTirocinio
        > DomandaTirocinio
        > DomandaTirocinioNonValidaException
        > DomandaTirocinioRepository
        > DomandaTirocinioService
      > progettoFormativo
      > responsabileUfficioTirocini
      > Studente
      > utenza
      > web
        > ProjectInitializer
        > Webapp
```



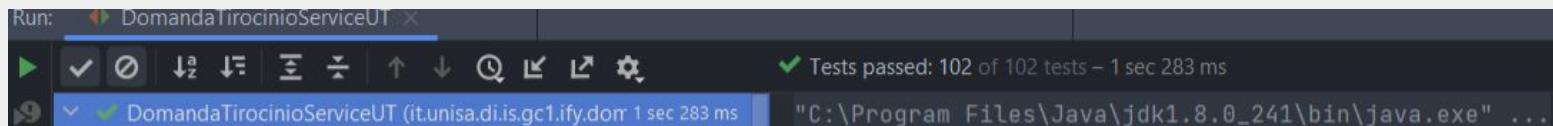
TESTING DI UNITÀ

Esempio caso di test:

```
@Test
public void accettaDomandaTirocinioGiaAccetataDalTutor() {
    domanda.setStato(DomandaTirocinio.IN_ATTESA_AZIENDA);
    when(utenzaService.getUtenteAutenticato()).thenReturn(delegate);
    when(domandaTirocinioRepository.findById(domanda.getId())).thenReturn(Optional.of(domanda));
    when(domandaTirocinioRepository.save(domanda)).thenReturn(domanda);
    try {
        domandaTirocinioService.accettaDomandaTirocinioByAzienda(domanda.getId());
    } catch (OperazioneNonAutorizzataException e) {
        e.printStackTrace();
    }
    verify(domandaTirocinioRepository, times(wantedNumberOfInvocations: 1)).save(domanda);
}
```

L'esecuzione dei test di unità non ha prodotto nessuna failure e l'approccio utilizzato ha garantito un'ispezione profonda della logica di business implementata nelle classi Service

Esecuzione di una classe di test:





TESTING DI INTEGRAZIONE

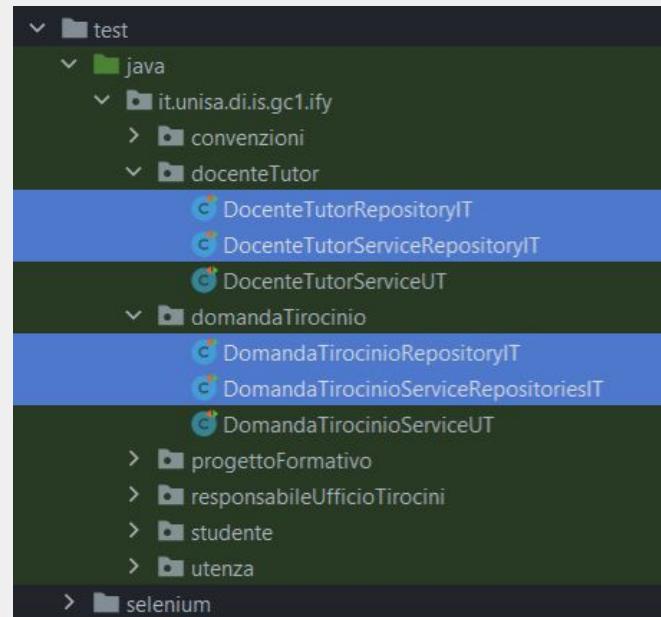
Approcci utilizzati

- Bottom-UP

Componenti Testate

- Classi Repository con la base dati
- Classi Service con classi repository

Sono state testate le componenti inerenti alle modifiche



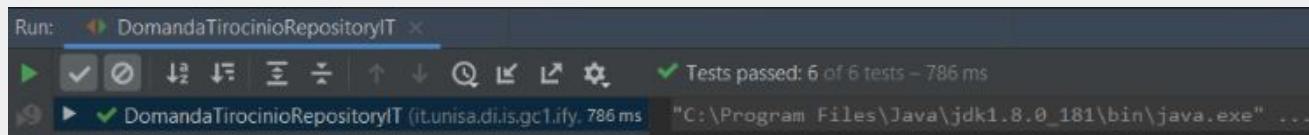


TESTING DI INTEGRAZIONE

Esempio testing d'integrazione

```
@Test
public void findAllByStudenteId() {
    for (Studente studente : listaStudenti) {
        List<DomandaTirocinio> domandeRestituite = domandeRepository.findAllByStudenteId(studente.getId());
        for (DomandaTirocinio domandaRestituita : domandeRestituite)
            assertThat(listaDomande.contains(domandaRestituita), is( value: true));
    }
}
```

Esecuzione di una classe di test:



L'esecuzione dei test di integrazione ha permesso di verificare l'assenza di comportamenti inaspettati nell'interazione tra le componenti.



COVERAGE DI TESTING RAGGIUNTA

Dettagli branch coverage raggiunta:

89% classes, 76% lines covered in 'all classes in scope'				
Element ▾	Class, %	Method, %	Line, %	Branch, %
it.unisa.di.is.gc1.ify	89% (62/69)	74% (588/794)	76% (5331/6959)	85% (572/666)

89% classes, 76% lines covered in package 'it.unisa.di.is.gc1.ify'				
Element ▾	Class, %	Method, %	Line, %	Branch, %
Webapp	100% (1/1)	0% (0/1)	33% (1/3)	100% (0/0)
ProjectInitializer	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
web	68% (13/19)	2% (5/174)	2% (29/984)	0% (0/82)
utenza	100% (9/9)	96% (53/55)	96% (370/385)	97% (45/46)
studente	100% (5/5)	98% (63/64)	89% (928/1032)	100% (10/10)
Studente	100% (5/5)	73% (30/41)	86% (150/173)	100% (90/90)
responsabileUfficioTirocini	100% (2/2)	100% (7/7)	100% (51/51)	100% (4/4)
progettoFormativo	100% (6/6)	95% (94/98)	84% (574/677)	100% (78/78)
domandaTirocinio	100% (6/6)	98% (188/191)	85% (1586/1855)	95% (214/224)
docenteTutor	100% (3/3)	100% (8/8)	100% (49/49)	100% (4/4)
DocenteTutor	100% (2/2)	71% (5/7)	68% (11/16)	100% (0/0)
convenzioni	100% (10/10)	91% (135/147)	91% (1582/1733)	99% (127/128)

Nel report sovrastante, si considera la coverage raggiunta con tutti i test cases, tra cui quelli eseguiti tramite testing di regressione.



TESTING DI SISTEMA

Il testing di sistema è servito per evidenziare eventuali errori che si sono verificati durante i test.

Approccio utilizzato - Category Partition:

utilizzato per testare i form di inserimento dati inerenti ai cambiamenti apportati, eseguito sulle possibili combinazioni di valori che può inserire l'utente. I form testati fanno riferimento alle funzionalità elencate.

Per ogni funzionalità testata:

- Definizione dei parametri della funzionalità
- Definizione delle categorie per ogni parametro

Parametro	Tipo	Numero caratteri
E-mail	String	256
Password	String	24

Parametro: Password	
Categorie	Scelte
Corrispondenza Password cp	1: La password non corrisponde [if formatoFEok] [corrispondenzapasswordCPko] 2: La password corrisponde [if formatoFEok] [corrispondenzapasswordCPok]

Funzionalità testate:

- Login
- Invio della domanda di tirocinio

Codice	Combinazione	Esito
TC_LP_1	fe1	Email non valida
TC_LP_2	fe2.ee1	Email non presente nel sistema
TC_LP_3	fe2.ee2.cp1	Password non valida
TC_LP_4	fe2.ee2.cp2	Login Effettuato



TESTING DI SISTEMA

Test Case ID	TC_LP_1
Pre-condition:	
L'ospite <u>richiede</u> di autenticarsi alla piattaforma. L' ospite è un docente pre-registrato alla piattaforma.	
Flow of events:	
1. l'ospite dalla pagina di login inserisce i dati:	
Input	Valore
Email	
Password	Password1
2. L'ospite clicca sul pulsante "Accedi"	
Oracle:	
Il sistema visualizza il messaggio: "Email non valida". Il login non viene effettuato.	

Test Case ID	TC_LP_2
Pre-condition:	
L'ospite <u>richiede</u> di autenticarsi alla piattaforma. L' ospite è un docente pre-registrato alla piattaforma.	
Flow of events:	
1. l'ospite dalla pagina di login inserisce i dati:	
Input	Valore
Email	p.verdi@unisa.it
Password	Password1
2. L'ospite clicca sul pulsante "Accedi"	
Oracle:	
Il sistema visualizza il messaggio: "Email non presente nel sistema". Il login non viene effettuato.	



TESTING DI SISTEMA

Esecuzione dei test case tramite selenium IDE

Test Case	Data e Ora Esecuzione	Pass/Fail
TC_LP_1	16/07/2021 - 15:40	Pass
TC_LP_2	16/07/2021 - 15:40	Pass
TC_LP_3	16/07/2021 - 15:40	Pass
TC_LP_4	16/07/2021 - 15:40	Pass

Running 'TC_LP_1'

1. open on <http://localhost:8080> OK
2. setWindowSize on 1413x904 OK
3. click on linkText=Login OK
4. click on id=inputEmail OK
5. click on id=inputPassword OK
6. type on id=inputPassword with value Password1 OK
7. click on id=reg OK
8. assertText on css=.myError with value Email non valida OK

'TC_LP_1' completed successfully

Running 'TC_LP_2'

1. open on <http://localhost:8080> OK
2. setWindowSize on 1413x904 OK
3. click on linkText=Login OK
4. click on id=inputEmail OK
5. type on id=inputEmail with value p.verdi@unisa.it OK
6. click on id=inputPassword OK
7. type on id=inputPassword with value Password1 OK
8. click on id=reg OK
9. assertText on css=.myError with value Email non presente nel sistema OK

'TC_LP_2' completed successfully

Grazie per l'ascolto