



Maintenance Report

IFY Internship For You

Riferimento

Versione 1.0

Data 29/04/2021

Destinatario Prof. Andrea De Lucia

Presentato da Carmine Ferrara, Francesco Abate e
Gianluca Spinelli

Sommario

1. Introduzione	3
2. Architettura del sistema attuale	5
2.1 Panoramica	5
2.2 Decomposizione in sottosistemi	5
2.2.1 Architettura del sistema e sottosistemi individuati	5
2.3 Gestione Dati Persistenti	7
3. Descrizione delle modifiche change request 1	8
3.1 Analisi della richiesta di modifica	8
3.2 Analisi d'impatto	8
3.3 Metriche di Impact Analysis	21
4. Descrizione delle modifiche change request 2	22
4.1 Analisi della richiesta di modifica	22
4.2 Analisi d'impatto	22
4.3 Metriche di Impact Analysis	28
5. Testing	29
5.1 Obiettivi e Criteri di Testing Plan	29
5.2 Metodologie e Livelli del testing	29
5.3 Specifiche Testing di sistema	30
5.4 Testing Execution Report	51

1. Introduzione

A seguito delle Change Request presentate al docente, il lavoro di manutenzione si è concentrato nello studio del dominio di analisi e progettazione del sistema attuale, in modo tale da individuare le componenti impattate dalla richiesta di modifica ed eventuali componenti aggiuntive per adattare il nuovo flusso operativo del sistema a seguito dell'introduzione del nuovo ruolo: Tutor accademico.

Data la prima change request, si evince la necessità di introdurre una nuova tipologia di utente all'interno del sistema, la quale dovrà occuparsi della supervisione e dell'approvazione della richiesta di tirocinio effettuata da parte dello studente.

In analogia a quelle che sono le modalità di approvazioni aziendali, da parte del tutor aziendale, il docente tutor dovrà avere le seguenti funzionalità:

- Possibilità di accesso alla propria area personale (con credenziali fornite direttamente dal team di manutenzione, come espresso nella change request);
- possibilità di visualizzare le richieste di tirocinio presentate da uno studente e non ancora approvate (in attesa oppure approvate solo dal tutor aziendale);
- possibilità di visualizzare le domande di tirocinio approvate (dal tutor e dall'azienda) e trasmesse all'ufficio carriere di dipartimento;
- possibilità di approvare o rifiutare una domanda di richiesta di uno studente;
- visualizzare i tirocini approvati in via definitiva, di cui esso è referente, i quali non sono ancora terminati;

L'aggiunta di queste funzionalità comporta anche un cambiamento al workflow della domanda di tirocinio e ai processi di business ad essa correlati, i quali appunto dovranno essere modificati e testati in modo opportuno.

Data la seconda change request, si evince la necessità di introdurre la possibilità, da parte del tutor accademico, di concludere l'esperienza di tirocinio dello studente con successo e quindi visualizzare nella propria area riservata, sia i tirocini terminati con successo sia quelli scaduti.

In relazione alla change request data, si evince una modifica del flusso relativo alla domanda di tirocinio, comprendente l'aggiunta di nuovi stati alla domanda di tirocinio e l'aggiunta di nuove funzionalità da parte del tutor accademico:

- Possibilità di terminare il tirocinio di uno studente con successo dalla propria area riservata;
- possibilità di visualizzare i tirocini scaduti e terminati con successo;

In questo documento vengono analizzate e documentate le due change request in termini di componenti impattate e ne viene documentato il testing.

2. Architettura del sistema attuale

Al fine di comprendere il funzionamento del sistema prima della modifica sono state riportate le principali caratteristiche della struttura.

Fonte: IFY_SDD_V1.0

2.1 Panoramica

Il sistema attuale prevede diverse funzionalità mirate a seconda della tipologia di utente (studente, delegato aziendale, responsabile ufficio tirocini) che ha effettuato l'accesso alla piattaforma, nel dettaglio:

- lo studente, partendo dalla visualizzazione dei progetti formativi di un'azienda, potrà inoltrare una domanda di tirocinio e seguirne lo stato di avanzamento dalla sua dashboard;
- Il delegato aziendale potrà inserire e gestire progetti formativi inerenti alle attività di tirocinio che l'azienda intende fornire. Inoltre, potrà accettare o rifiutare le domande di tirocinio pervenute dagli studenti e successivamente visualizzarne l'evoluzione;
- Il responsabile dell'ufficio tirocini potrà visualizzare le domande di tirocinio pervenute dopo l'accettazione aziendale, gestirne la valutazione finale.

Ogni tipologia di utente registrato avrà, inoltre, una visualizzazione mirata al ruolo ricoperto nel sistema dei tirocini in corso.

2.2 Decomposizione in sottosistemi

2.2.1 Architettura del sistema e sottosistemi individuati

Come specificato nella definizione teorica del pattern MVC, la decomposizione del nostro sistema è composta da tre layer aventi ciascuna aspetti e funzionalità differenti:

- View: layer di presentazione delle funzionalità del sistema e relativo interfacciamento con l'utenza;
- Controller: layer che si occupa della gestione della logica di business del sistema e relativi flussi di controllo dei dati;
- Model: layer che si occupa della gestione dei dati persistenti e relativo interscambio degli stessi tra i sottosistemi.

Oggetti Boundary individuati nella fase di analisi definiamo un sottosistema view:

- **UIterface** (sottosistema per la gestione dell'interfacciamento con l'utente).

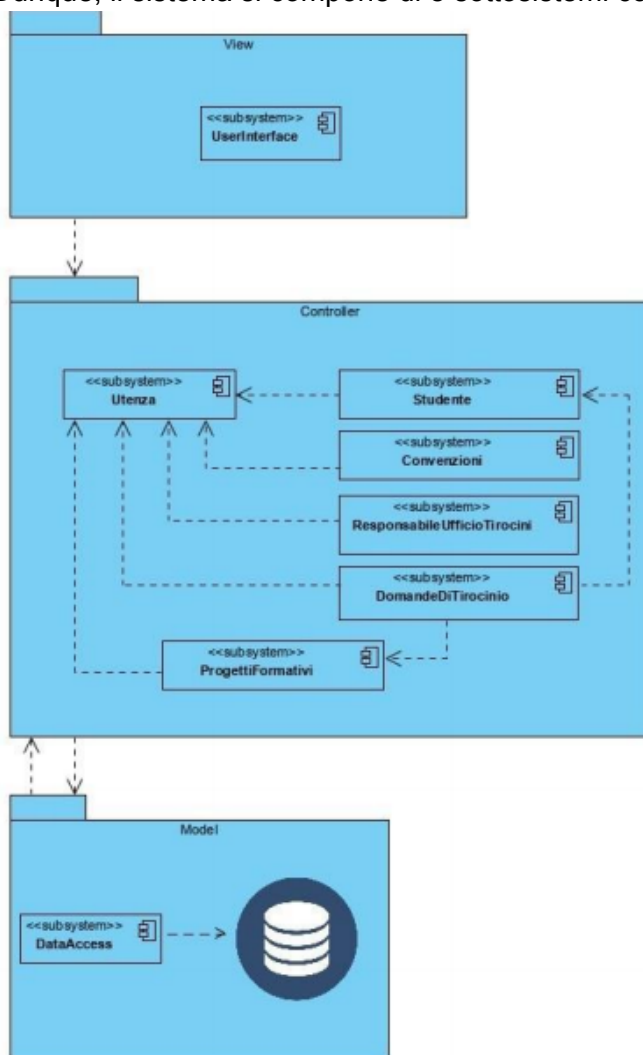
Oggetti Controller individuati nella fase di analisi definiamo sei sottosistemi:

- **Utenza** (sottosistema per la gestione del login, logout degli utenti registrati; delle richieste di registrazione);
- **Responsabile** (sottosistema per la gestione dei dati del responsabile);
- **Studiante** (sottosistema per la gestione delle richieste di registrazione);
- **Convenzioni** (sottosistema per la gestione delle convenzioni);
- **DomandeDiTirocinio** (sottosistema per la gestione delle domande di tirocinio);
- **ProgettiFormativi** (sottosistema per la gestione dei progetti formativi).

Da un'analisi degli oggetti Model individuati nella fase di analisi definiamo un sottosistema:

- **DataAccess**.

Dunque, il sistema si compone di 8 sottosistemi come mostrato in figura:



2.3 Gestione Dati Persistenti

Il sistema attuale gestisce la memorizzazione di dati persistenti tramite l'utilizzo di un database relazionale per consentire un accesso veloce e sicuro ai dati e un ampio spazio di archiviazione. Grazie all'uso del DBMS i dati sono protetti quindi diversi utenti con differenti operazioni possono accedere a diverse sezioni del database. Viene anche garantito l'accesso concorrente ai dati affidabili salvandone una copia in modo da poter ripristinare lo stato del database in caso di danni software o hardware.

I dati persistenti sono organizzati nel seguente modo:



// Si ricorda che la domanda tirocinio include anche il concetto logico di tirocinio in sé, in quanto tiene traccia dello stato in cui la domanda di tirocinio e quindi il tirocinio in sé potrebbe trovarsi. Quindi, le funzionalità relative ai tirocini sono state implementate durante il processo di sviluppo sull'entità persistente "Domanda tirocinio". Un esempio è proprio la visualizzazione dei tirocini in corso che il sistema mette a disposizione. //

[Fonte IFY_RADv1.1]

3. Descrizione delle modifiche change request 1

La change request 1 richiede una manutenzione evolutiva, in quanto, introduce un nuovo attore all'interno del sistema: "Tutor Accademico" ed aggiunge nuove funzionalità relative ad esso. Per maggiori dettagli consultare il documento "[IFY Change Requestes.docx](#)".

3.1 Analisi della richiesta di modifica

La prima richiesta di modifica introduce quindi un nuovo attore all'interno del sistema che agirà in maniera diretta sulle richieste di tirocinio inoltrate dagli studenti le quali, dovranno quindi, essere approvate in prima fase sia dal tutor accademico che dal tutor aziendale e solo successivamente dal responsabile dell'ufficio didattico, per poter essere avviate.

Quindi, a differenza dell'attuale workflow, le richieste di tirocinio dovranno essere adattate in maniera tale da coinvolgere il tutor accademico nel processo di richiesta del tirocinio. Inoltre, non essendo presente l'area riservata inerente al tutor accademico, si dovrà procedere con:

- Realizzazione dell'interfaccia dedicata al tutor accademico;
- visualizzare le richieste in attesa della sua approvazione;
- visualizzare le richieste da lui approvate in attesa dell'approvazione aziendale o d'ufficio oppure definitivamente approvate;
- possibilità di accettare o rifiutare una richiesta di tirocinio per incongruenza;
- visualizzare i tirocini approvati in corso di svolgimento, così come già possibile per il tutor aziendale, lo studente e l'ufficio.

Volendo quindi aggiungere queste nuove funzionalità si rende necessario analizzare le componenti che saranno impattate dalla suddetta modifica e valutarne quanto impatta. Si procederà con l'analisi degli artefatti impattati seguendo un approccio top-down, quindi partendo dalla documentazione fino ad arrivare alle singole classi.

3.2 Analisi d'impatto

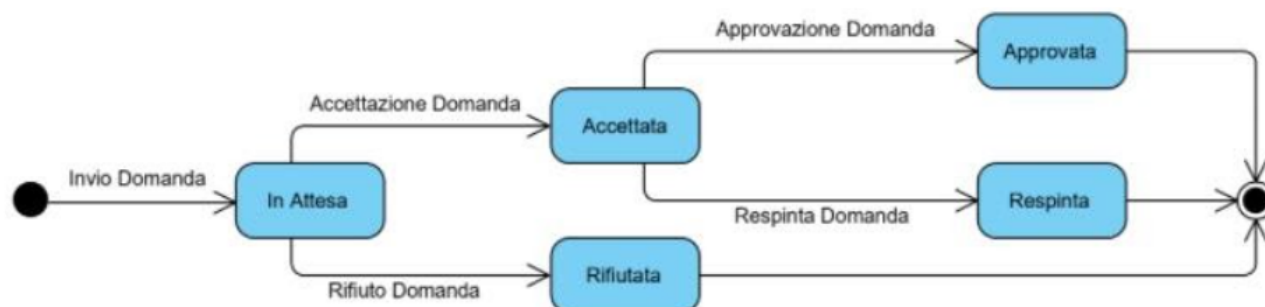
Per implementare le modifiche al sistema, il lavoro è partito con lo studio del workflow della domanda di tirocinio nel dominio di analisi, in particolare il team di manutenzione si è concentrato sugli artefatti inerenti alla domanda di tirocinio.

Passo 1:

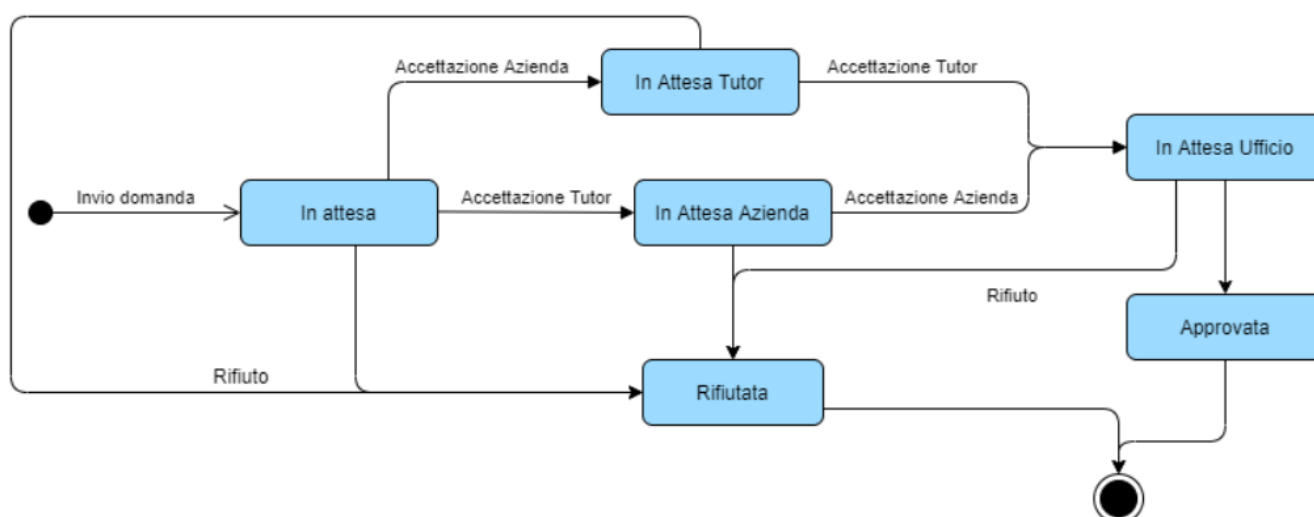
E' stata applicata una modifica allo statechart inerente la domanda di tirocinio, in modo tale da avere una chiara visione degli stati e degli eventi correlati alla domanda di tirocinio a seguito delle attività di modifica.

Statechart prima della modifica:

Fonte: IFY_RAD_V1.1



Statechart dopo la modifica:



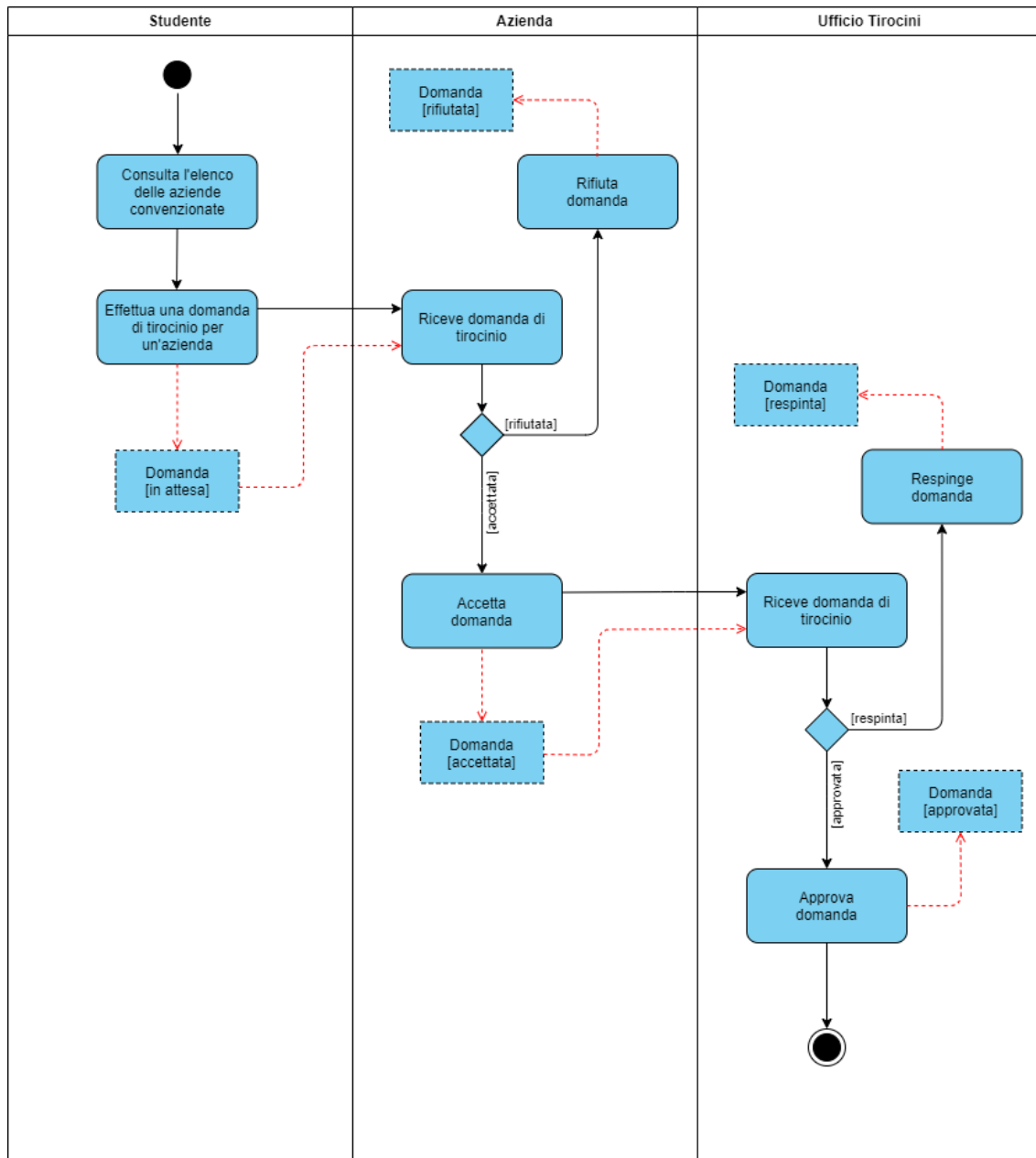
Dal diagramma modificato si prevedono più stati per la richiesta di tirocinio in modo tale che essa possa essere approvata in maniera indipendente dal tutor accademico e dal tutor aziendale e solo a seguito di entrambe le approvazioni, potrà essere sottoposta all'ufficio di segreteria per la convalida finale.

Passo 2:

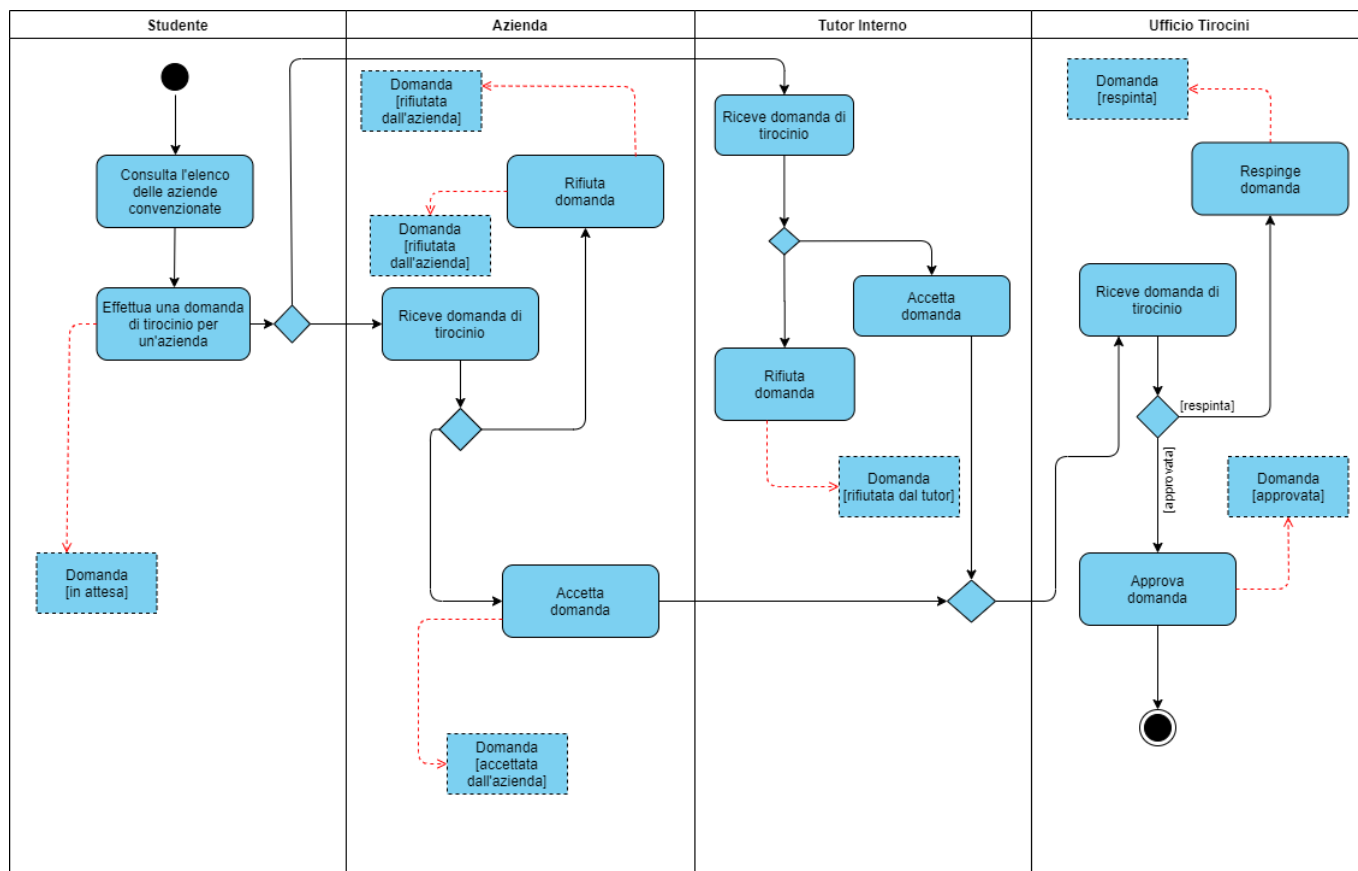
Avendo quindi individuato questi nuovi stati, il lavoro è proseguito con l'analisi delle attività che i singoli attori svolgono all'interno del processo di business inerente alla domanda di tirocinio.

Activity Diagram prima:

Fonte: IFY_RAD_V1.1



Activity Diagram dopo:



La modifica sostanziale è stata introdurre una nuova swimlane: Tutor Interno, che contiene la sequenza di attività relative all'accettazione o rifiuto della domanda di tirocinio.

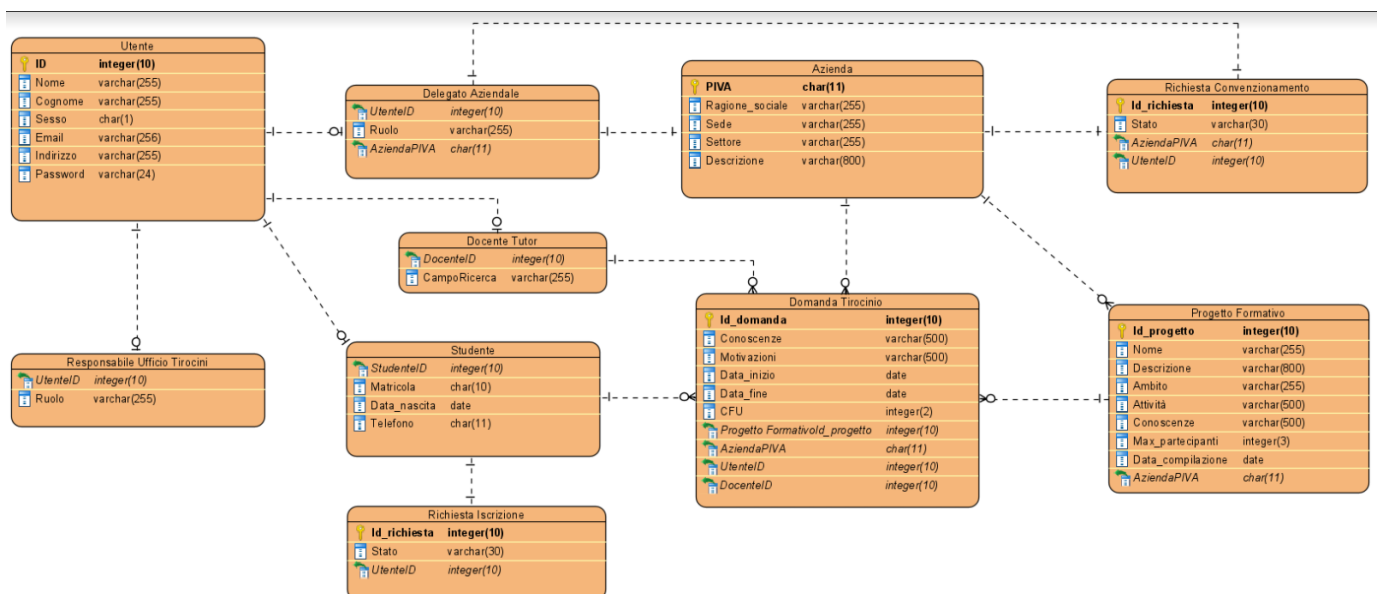
Passo 3:

Dopo aver apportato le modifiche agli statechart e agli activity diagram, il lavoro è proseguito con l'analisi dell'architettura attuale, partendo dalla strutturazione dei dati persistenti. Dopo aver analizzato il diagramma è stata introdotta una nuova specializzazione, "*Docente Tutor*", ed è stata inserita la chiave esterna nella relazione "*Domanda Tirocinio*".

Fonte: IFY_SDD_V1.0



DBD dopo:



Passo 4: Avendo analizzato la documentazione iniziale ed aver modificato i diagrammi più rilevanti per la richiesta di modifica, si è quindi deciso di redarre lo starting impact set in modo tale da poter comprendere cosa modificare per soddisfare la prima change request ad una granularità più alta.

/*Si specifica che lo starting impact set, è stato realizzato con una granularità di classe, in quanto la modifica non solo va ad introdurre nuovi metodi, ma andrà ad impattare metodi già presenti derivanti dal precedente progetto di sviluppo. Con la successiva definizione del Candidate Impact Set si è quindi deciso di adottare un livello di astrazione più basso, considerando i singoli attributi e metodi aggiunti o modificati, avendo a disposizione più dettagli, data l'analisi dei Ripple Effects effettuata tramite il diagramma delle classi*/

STARTING IMPACT SET

- **DocenteTutor** -> creazione della classe bean “DocenteTutor” con getter, setter e pattern per la validazione;
- **DocenteTutorRepository** -> creazione dell'interfaccia “DocenteTutorRepository” composta da firme rappresentanti le query al database inerenti al docente tutor;
- **DocenteTutorService** -> creazione della classe “DocenteTutorService” composta da funzioni che fanno utilizzo delle funzioni di DocenteTutorRepository ai fini della logica applicativa;
- **DomandaTirocinio** -> aggiunta dell'attributo “tutor: DocenteTutor” e i relativi get e set, e nuovi stati relativi al nuovo flusso di approvazione che coinvolge il tutor;
- **DomandaTirocinioService** -> aggiunta dei seguenti metodi per le funzionalità utente:

```
accettaDomandaTirocinioByDocente(id : Long) : DomandaTirocinio  
rifiutaDomandaTirocinioByDocente(id : Long) : DomandaTirocinio  
visualizzaDomandeTirocinioInAttesaDocente(idDocente : Long) : List<DomandaTirocinio>  
visualizzaDomandeTirocinioInoltrateDocente(idDocente : Long) : List<DomandaTirocinio>  
visualizzaTirocinioInCorsoDocente(idDocente : Long) : List<DomandaTirocinio>  
validaDocenteTutor(docenteTutorId: String): String
```

Modifiche ai metodi relativi alla logica di Business Aziendale, data l'aggiunta di stati alla domanda di tirocinio (non si danno dettagli maggiori sui metodi impattati, tale operazione, verrà effettuata nel candidate impact set a seguito dell'analisi dei ripple effects sul class diagram);

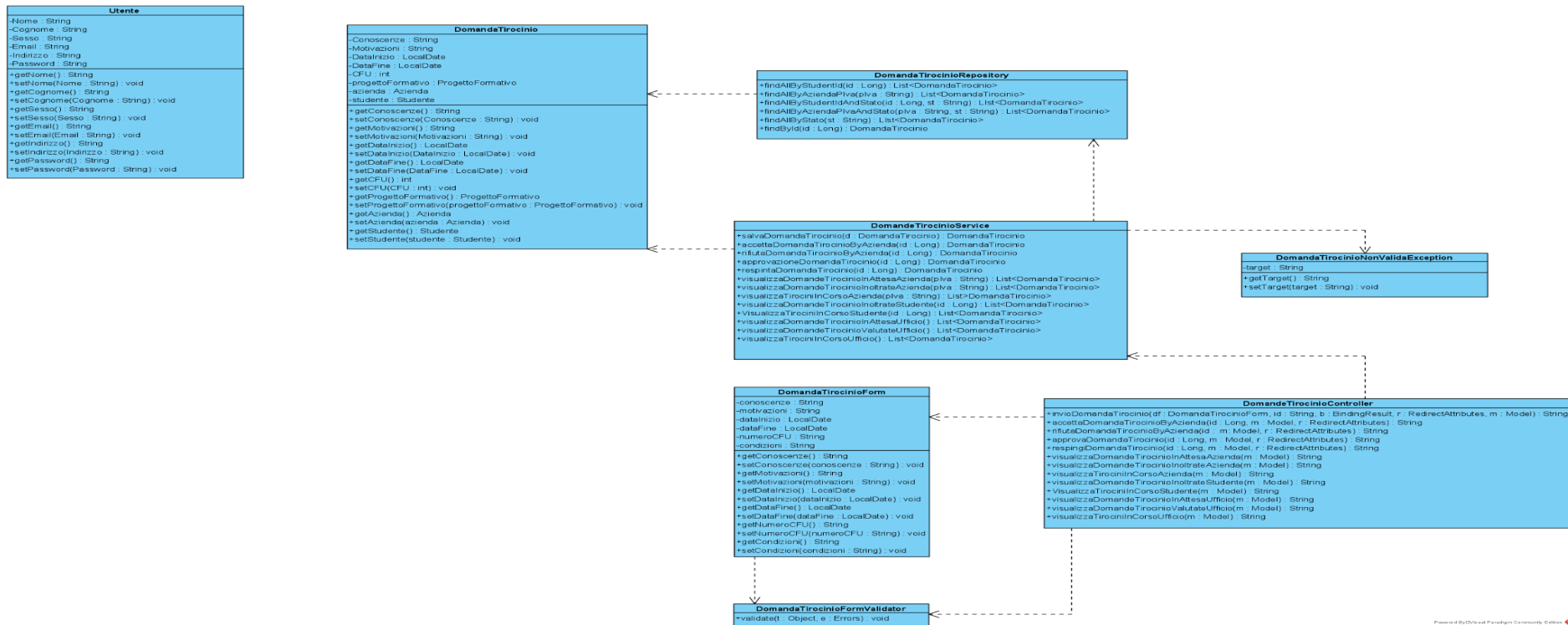
- **DomandaTirocinioController** -> aggiunti i seguenti metodi:

```
accettaDomandaTirocinioByDocente(id : Long) : DomandaTirocinio  
rifiutaDomandaTirocinioByDocente(id : Long) : DomandaTirocinio  
visualizzaDomandeTirocinioInAttesaDocente(idDocente : Long) : List<DomandaTirocinio>  
visualizzaDomandeTirocinioInoltrateDocente(idDocente : Long) : List<DomandaTirocinio>  
visualizzaTirocinioInCorsoDocente(idDocente : Long) : List<DomandaTirocinio>
```

Modifiche ai metodi relativi alla logica di Business Aziendale, data l'aggiunta di stati alla domanda di tirocinio (non si danno dettagli maggiori sui metodi impattati, tale operazione, verrà effettuata nel candidate impact set a seguito dell'analisi dei ripple effects sul class diagram);

- **DomandaTirocinioForm** -> aggiunto l'attributo “docenteEmail: String” con i relativi get e set.

Passo 5: Successivamente il lavoro è proseguito con l'analisi e la modifica del Class Diagram. Dal quale sono state estratte le seguenti classi e relative dipendenze che hanno impatto e devono essere modificate.





A seguito della ristrutturazione del Class Diagram, le componenti rilevate come membri del candidate impact set sono le seguenti:

CANDIDATE IMPACT SET - 55 componenti impattate

- **DocenteTutor** -> creazione della classe bean “DocenteTutor” con getter, setter e pattern per la validazione;
 - Si prevedono gli attributi nome, cognome, sesso, email, password, indirizzo, campoRicerca, quindi un totale di **7 attributi**;
 - Ogni attributo sarà accompagnato da un getter e da un setter per un totale di **14 metodi**;
 - Si prevedono **3 pattern** per la validazione del campo di ricerca: un pattern che regoli la formattazione del campo di ricerca, un pattern che indichi la lunghezza minima del campo e un pattern che indichi la lunghezza massima.
- **DocenteTutorRepository**-> creazione dell’interfaccia “DocenteTutorRepository” composta da firme rappresentanti le query al database inerenti al docente:
 - Si prevede la definizione di **un metodo** di ricerca per il database.
- **DocenteTutorService** -> creazione della classe “DocenteTutorService” composta da funzioni che fanno utilizzo delle funzioni di DocenteTutorRepository ai fini della logica applicativa:
 - Si prevede l’aggiunta di **2 metodi**: uno per ottenere tutti i docenti disponibili e uno per ottenere il docente corrispondente ad un determinato ID.
- **DomandaTirocinio** -> aggiunta dell’attributo “tutor: DocenteTutor” e i relativi get e set, e nuovi stati relativi al nuovo flusso di approvazione che coinvolge il tutor:
 - Si prevede l’aggiunta di **un attributo** “tutor: DocenteTutor”;
 - Si prevede l’aggiunta di getter e setter legati all’attributo “tutor”, quindi **2 metodi**;
 - Si prevede l’aggiunta di **2 stati** i quali fanno riferimento all’accettazione della domanda di tirocinio da parte del docente e del tutor aziendale;
 - Si prevede la rimozione di **uno stato**, precisamente “Accettata”, siccome è stato specializzato nei due stati aggiunti.
- **DomandaTirocinioRepository** ->
 - Si prevede l’aggiunta di **2 metodi** inerenti alla ricerca delle domande di tirocinio tramite l’ID del docente;
- **DomandaTirocinioService** ->
 - Si prevede l’aggiunta di **5 metodi** inerenti all’accettazione, rifiuto e visualizzazione delle domande di tirocinio tramite l’ID del docente;
 - Si prevede la modifica di **4 metodi** di gestione dei tirocini in modo da rendere compatibili le approvazioni, rifiuti e visualizzazioni da parte dell’azienda all’aggiunta dei metodi inerenti al docente.
- **DomandaTirocinioController** ->
 - Si prevede l’aggiunta di **5 metodi** inerenti all’accettazione, rifiuto e visualizzazione delle domande di tirocinio tramite l’ID del docente;
 - Si prevede la modifica di **2 metodi** di gestione dei tirocini in modo da rendere compatibili le approvazioni e i rifiuti da parte dell’azienda all’aggiunta dei metodi inerenti al docente.
- **DomandaTirocinioForm** -> aggiunto l’attributo “docenteTutorId: String” con i relativi get e set:
 - Si prevede **un attributo** aggiuntivo nel form;
 - L’attributo sarà accompagnato da un getter e da un setter per un totale di **2 metodi**;
- **UtenzaController** -> specializzazione del metodo di login(...) in funzione del nuovo utente

DocenteTutor, senza intaccare la firma del metodo:

- Si prevede la modifica di **un metodo**, precisamente del metodo di login.

Segue l'elenco delle JSP che risultano essere candidate per essere aggiunte o modificate, le quali sono state individuate a seguito dell'analisi dei Ripple Effects:

- **docenteDashboard.jsp**
- **modallnvioDomandeTirocinio.jsp**
- **visualizzaDomandeTirociniInAttesaDocente.jsp**
- **visualizzaDomandeTirociniInoltrateDocente.jsp**
- **visualizzaDomandeTirociniInCorsoDocente.jsp**
- **Delegato Dashboard.jsp**
- **visualizzaDomandeTirociniInAttesaAzienda.jsp**

Passo 4: implementazione

A seguito dell'individuazione del Candidate Impact Set, si è proceduto con l'implementazione della modifica, Si riportano per ogni sessione di lavoro i dettagli delle singole componenti implementate.

Sessione di lavoro #1

Componenti aggiunte

Backend

- **DocenteTutor** -> crea la classe bean "DocenteTutor" con getter, setter e pattern per la validazione;
- **DocenteTutorService** -> crea la classe "DocenteTutorService" composta da funzioni che facciano da wrapper per le funzioni di DocenteTutorRepository;
- **DocenteTutorRepository**-> crea l'interfaccia "DocenteTutorRepository" composta da firme rappresentanti le query al database inerenti al docente tutor;

Frontend

- **docenteDashboard.jsp** -> crea la pagina dashboard per il docente tutor;

Componenti impattate

Backend

- **UtenzaController** -> aggiunto il controllo per la casistica DocenteTutor nel metodo di login;

Corso di *Ingegneria, Gestione ed Evoluzione del Software*

- **DomandaTirocinioFormValidator** -> eseguita la validazione del docente tutor tramite apposito service;
- **DomandaTirocinioForm** -> il form conserverà l'id del docente tutor scelto;
- **DomandaTirocinioController** -> i docenti disponibili vengono ora iniettati nel form per poter essere scelti dagli studenti
- **DomandaTirocinioService** -> aggiunta la funzione validaDocenteTutor(id) per controllare l'esistenza del docente sul database;
- **DomandaTirocinioRepository** -> aggiunte firme nell'interfaccia rappresentanti le query al database inerenti al docente tutor relazionato alla domanda di tirocinio;
- **DomandaTirocinio** -> la domanda di tirocinio conserverà l'id del docente tutor scelto; aggiunti nuovi stati per gestire l'approvazione e/o il rifiuto della richiesta da parte del docente tutor;

Frontend

- **modallnvioDomandeTirocinio.jsp** -> aggiunta la selezione del docente tutor quando si crea la richiesta per il tirocinio.

Sessione di lavoro #2

Componenti impattate

Backend

- **DomandaTirocinioService** -> filtrati gli stati da visualizzare tenendo conto dello stato di accettazione del docente tutor; metod refactoring;

accettaDomandaTirocinio -> accettaDomandaTirocinioByAzienda con modifiche sui controlli e settaggi dello stato della richiesta

rifiutoDomandaTirocinio -> rifiutoDomandaTirocinioByAzienda con modifiche sui controlli e settaggi dello stato della richiesta

visualizzaDomandeTirociniInoltrateAzienda : modifica del valore di ritorno, comprendente anche delle domande in attesa di accettazione della sola azienda.

visualizzaDomandeTirociniInAttesaAzienda: modifica del valore di ritorno, comprendente anche delle domande in attesa di accettazione della sola azienda.

- **DomandaTirocinioController** -> modifiche di adattamento ai metodi service precedentemente elencati
- **DomandaTirocinio** -> cambiati gli stati secondo le scelte della documentazione di manutenzione;
- **Mail Singleton Sender** -> generalizzati due messaggi data l'aggiunta dell'utente docente;

Frontend

- **visualizzaDomandeTirociniInAttesaAzienda.jsp** -> modificati i nomi di mapping dei relativi metodi di request del controller.

Sessione di lavoro #3

Componenti aggiunte

Frontend

- **visualizzaDomandeTirociniInAttesaDocente.jsp** -> aggiunta interfaccia per la visualizzazione delle domande di tirocinio in attesa di approvazione del docente;
- **visualizzaDomandeTirociniInoltrateDocente.jsp** -> aggiunta interfaccia per la visualizzazione delle domande di tirocinio approvate dal docente;
- **visualizzaTirociniiInCorsoDocente.jsp** -> aggiunta interfaccia per la visualizzazione dei tirocini in corso dal punto di vista del docente;

Componenti impattate

Backend

- **DomandaTirocinioService** -> fix di permessi errati;
- **UtenzaController** -> redirectione alla dashboard del docente in caso di eccezioni nei metodi inerenti al docente o in caso di pressione su qualsiasi pulsante che redirectioni alla homepage da parte del docente;
- **Mail Singleton Sender** -> cambiati i messaggi adattandoli in base agli stati stabiliti.

Frontend

- **docenteDashboard.jsp** -> modificata l'interfaccia grafica;
- **visualizzaDomandeTirociniInoltrateAzienda.jsp** -> trattato lo stato inerente all'approvazione del docente;

ACTUAL IMPACT SET - 58 componenti impattate

- **DocenteTutor** -> creazione della classe bean "DocenteTutor" con getter, setter e pattern per la validazione;
 - Si prevedono gli attributi nome, cognome, sesso, email, password, indirizzo, campoRicerca, quindi un totale di **7 attributi**;
 - Ogni attributo sarà accompagnato da un getter e da un setter per un totale di **14 metodi**;
 - Si prevedono **3 pattern** per la validazione del campo di ricerca: un pattern che regoli la formattazione del campo di ricerca, un pattern che indichi la lunghezza minima del campo e un pattern che indichi la lunghezza massima.
- **DocenteTutorRepository** -> creazione dell'interfaccia "DocenteTutorRepository" composta da firme rappresentanti le query al database inerenti al docente:
 - Si prevede la definizione di **un metodo** di ricerca per il database.

Corso di *Ingegneria, Gestione ed Evoluzione del Software*

- **DocenteTutorService** -> creazione della classe “DocenteTutorService” composta da funzioni che fanno utilizzo delle funzioni di DocenteTutorRepository ai fini della logica applicativa:
 - Si prevede l’aggiunta di **2 metodi**: uno per ottenere tutti i docenti disponibili e uno per ottenere il docente corrispondente ad un determinato ID.
- **DomandaTirocinio** -> aggiunta dell’attributo “tutor: DocenteTutor” e i relativi get e set, e nuovi stati relativi al nuovo flusso di approvazione che coinvolge il tutor:
 - Si prevede l’aggiunta di **un attributo** “tutor: DocenteTutor”;
 - Si prevede l’aggiunta di getter e setter legati all’attributo “tutor”, quindi **2 metodi**;
 - Si prevede l’aggiunta di **2 stati** i quali fanno riferimento all’accettazione della domanda di tirocinio da parte del docente e del tutor aziendale;
 - Si prevede la rimozione di **uno stato**, precisamente “Accettata”, siccome è stato specializzato nei due stati aggiunti.
- **DomandaTirocinioRepository** ->
 - Si prevede l’aggiunta di **2 metodi** inerenti alla ricerca delle domande di tirocinio tramite l’ID del docente;
- **DomandaTirocinioService** ->
 - Si prevede l’aggiunta di **5 metodi** inerenti all’accettazione, rifiuto e visualizzazione delle domande di tirocinio tramite l’ID del docente;
 - Si prevede la modifica di **4 metodi** di gestione dei tirocini in modo da rendere compatibili le approvazioni, rifiuti e visualizzazioni da parte dell’azienda all’aggiunta dei metodi inerenti al docente.
- **DomandaTirocinioController** ->
 - Si prevede l’aggiunta di **5 metodi** inerenti all’accettazione, rifiuto e visualizzazione delle domande di tirocinio tramite l’ID del docente;
 - Si prevede la modifica di **2 metodi** di gestione dei tirocini in modo da rendere compatibili le approvazioni e i rifiuti da parte dell’azienda all’aggiunta dei metodi inerenti al docente.
- **DomandaTirocinioForm** -> aggiunto l’attributo “docenteTutorId: String” con i relativi get e set:
 - Si prevede **un attributo** aggiuntivo nel form;
 - L’attributo sarà accompagnato da un getter e da un setter per un totale di **2 metodi**;
- **UtenzaController** -> specializzazione del metodo di login(...) in funzione del nuovo utente DocenteTutor, senza intaccare la firma del metodo **e specializzazione del metodo di redirectione alla homepage**:
 - Si prevede la modifica di **due metodi**, precisamente del metodo di login e del metodo di redirectione alla homepage.
- **MailSingletonSender** -> modifica di **un metodo**, precisamente del **metodo per inviare la mail** al fine di prevedere messaggi che trattino i nuovi stati introdotti dall’aggiunta del docente.

Segue l’elenco delle JSP che risultano essere state effettivamente modificate o aggiunte durante la fase implementativa:

- **docenteDashboard.jsp**
- **modallnvioDomandeTirocinio.jsp**
- **visualizzaDomandeTirociniInAttesaDocente.jsp**
- **visualizzaDomandeTirociniInoltreDocente.jsp**
- **visualizzaDomandeTirociniInCorsoDocente.jsp**
- **Delegato Dashboard.jsp**
- **visualizzaDomandeTirociniInAttesaAzienda.jsp**

- [visualizzaDomandeTirociniInoltrateAzienda.jsp](#)

A seguito dell'implementazione non sono stati trovati falsi positivi, ma come si può ben notare dall'actual impact set sono state rilevate 3 componenti non incluse nel candidate impact set (componenti evidenziate in blu precedentemente).

Summary degli insiemi evinti dall'analisi d'impatto:

CANDIDATE IMPACT SET - 55 componenti impattate

ACTUAL IMPACT SET - 58 componenti impattate

FALSE POSITIVE IMPACT SET - 0 componenti impattate

DISCOVERING POSITIVE IMPACT SET - 3 componenti impattate

3.3 Metriche di Impact Analysis

A seguito delle operazioni di modifica, e avendo individuato gli insiemi di componenti relativi all'analisi d'impatto sono state calcolate ReCall, Precision, Adequacy ed Effectiveness, al fine di valutare la bontà dell'analisi d'impatto effettuata:

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

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

Inclusiveness = 0

Non avendo quindi individuato tutte le componenti effettivamente impattate dalla richiesta di modifica, si ottiene un Inclusiveness pari a 0, quindi teoricamente, l'analisi di impatto effettuata non ricopre una situazione ideale di adeguatezza, dato che l'AIS non è un sottoinsieme del CIS, se però si considerano le metriche di Information Retrieval per l'impact analysis, si osserva che i valori (di precision e recall per l'appunto) sono molto vicini ad 1 quindi quasi nella situazione ideale per una richiesta di modifica.

È comunque importante ricordare che la granularità degli elementi è molto fine, e che comunque in una situazione reale, difficilmente precision e recall sono effettivamente pari ad 1.

4. Descrizione delle modifiche change request 2

La change request 2 richiede una manutenzione evolutiva, in quanto, introduce al tutor accademico la possibilità di terminare con successo un tirocinio. Per maggiori dettagli consultare il documento "[IFY_Change Requestes.docx](#)".

4.1 Analisi della richiesta di modifica

La seconda richiesta di modifica introduce quindi un nuovo requisito funzionale relativo al tutor accademico, ovvero la possibilità di chiudere un'esperienza di tirocinio di uno studente, una volta che questi ha terminato l'esperienza. Questa operazione viene effettuata del tutor accademico all'interno della propria area riservata, nella sezione "Tirocini in corso", all'interno della quale egli potrà, oltre a visualizzare i tirocini in corso ed eventualmente terminarli, visualizzare anche i tirocini conclusi sia con successo che scaduti.

Quindi, a differenza dell'attuale workflow, il tutor accademico una volta approvato il tirocinio avrà la possibilità di terminarlo, prima della data di scadenza. Inoltre, da un punto di vista di stato, si ha l'introduzione di un nuovo stato "terminato" che indica un tirocinio concluso (con successo o non). Pertanto, la change request in questione richiede di:

- Modificare l'interfaccia dedicata alla visualizzazione dei tirocini in corso del tutor accademico per permettere la conclusione di questi;
- introduzione di un nuovo stato in cui si può trovare la domanda di tirocinio;
- visualizzare dei tirocini conclusi, per terminazione del docente e per scadenza pratica;

Volendo quindi aggiungere queste nuove funzionalità si rende necessario analizzare le componenti che saranno impattate dalla suddetta modifica e valutarne quanto impatta. Si procederà con l'analisi degli artefatti impattati seguendo un approccio top-down, quindi partendo dalla documentazione fino ad arrivare alle singole classi.

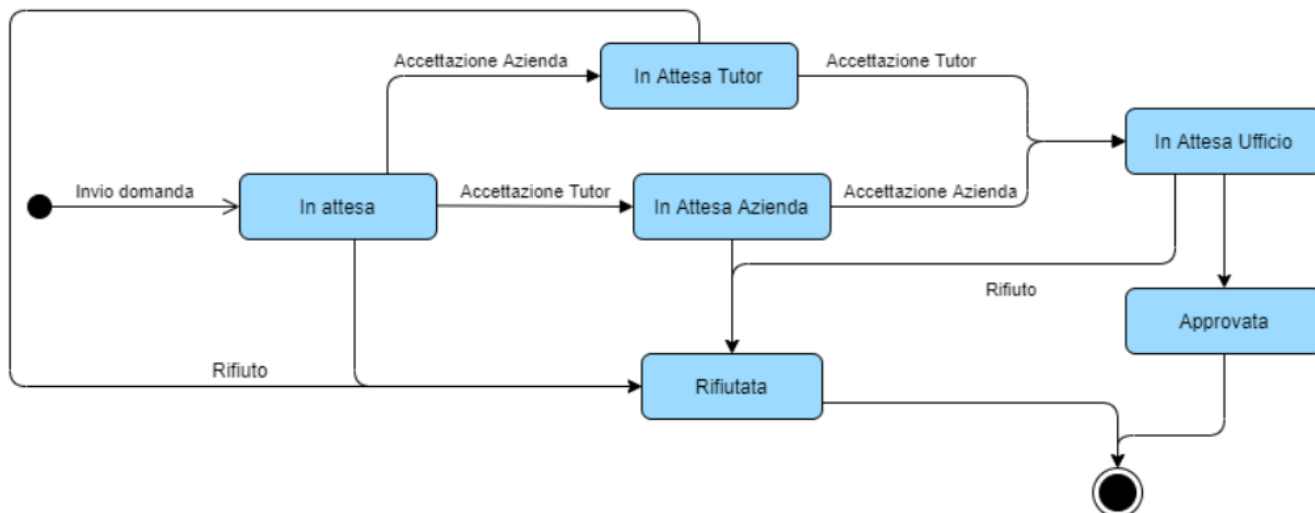
4.2 Analisi d'impatto

Per implementare le modifiche al sistema, il lavoro è partito con lo studio del workflow della domanda di tirocinio nel dominio di analisi, in particolare il team di manutenzione si è concentrato sugli artefatti inerenti alla domanda di tirocinio.

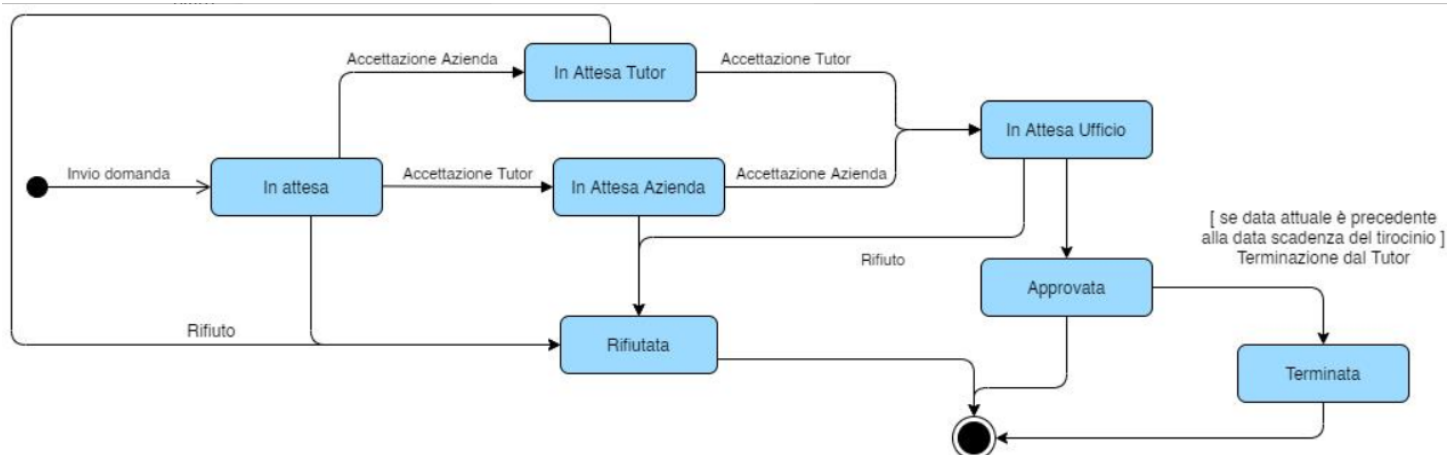
Passo 1:

E' stata applicata una modifica allo statechart inerente la domanda di tirocinio, in modo tale da avere una chiara visione degli stati e degli eventi correlati alla domanda di tirocinio a seguito delle attività di modifica.

Statechart prima della modifica:



Statechart dopo la modifica:



Dal diagramma modificato si prevede un nuovo stato per la richiesta di tirocinio in modo tale che essa possa essere terminata con successo dal tutor accademico.

Passo 2:Avendo analizzato la documentazione iniziale ed aver modificato i diagrammi più rilevanti per la richiesta di modifica, si è quindi deciso di redarre lo starting impact set in modo tale da poter poter comprendere cosa modificare per soddisfare la prima change request ad una granularità più alta.

/*Si specifica che lo starting impact set, è stato realizzato con una granularità di classe, in quanto la modifica non solo va ad introdurre nuovi metodi, ma andrà ad impattare metodi già presenti derivanti dal precedente progetto di sviluppo. Con la successiva definizione del Candidate Impact Set si è quindi deciso di adottare un livello di astrazione più basso, considerando i singoli attributi e metodi aggiunti o modificati, avendo a disposizione più dettagli, data l'analisi dei Ripple Effects effettuata tramite il diagramma delle classi*/

STARTING IMPACT SET

DomandaTirocinio -> aggiunta nuovo stato = TERMINATA.

DomandaTirocinioController -> aggiunta metodo Post terminaTirocinio.

DomandaTirocinioService -> aggiunta metodo terminaTirocinio. All'interno di questo metodo avviene il passaggio di stato del tirocinio da APPROVATA a TERMINATA e viene modificata la data di termine del tirocinio con la data odierna.

visualizzaTirociniCorsoDocente.jsp -> inserito pulsante per terminare il tirocinio. Inserita colonna "Data fine" e colonna "Stato". Inserita tabella dei tirocini conclusi contenente sia i tirocini terminati con successo, sia i tirocini scaduti. Questi ultimi vengono visualizzati sulla base del controllo tra la data odierna e la data di scadenza, pertanto non avviene una modifica all'interno del db (per semplicità si è scelta la sola modifica front-end).

Passo 3:

Successivamente il lavoro è proseguito con l'analisi e la modifica del Class Diagram. Dal quale sono state estratte le seguenti classi e relative dipendenze che hanno impatto e devono essere modificate.





A seguito della ristrutturazione del Class Diagram, le componenti rilevate come membri del candidate impact set sono le seguenti:

CANDIDATE IMPACT SET - 6 componenti impattate

DomandaTirocinio -> prevista l'aggiunta del **nuovo stato** = TERMINATA.

DomandaTirocinioController -> prevista l'aggiunta di **1 metodo** Post terminaTirocinio.

DomandaTirocinioService -> prevista l'aggiunta di **2 metodi** terminaTirocinio e visualizzaTirociniChiusi. All'interno di questo metodo avviene il passaggio di stato del tirocinio da APPROVATA a TERMINATA e viene modificata la data di termine del tirocinio con la data odierna.

MailSingletonSender -> modifica di **un metodo**, precisamente del **metodo per inviare la mail** al fine di prevedere messaggi che trattino i nuovi stati introdotti dall'aggiunta del docente.

visualizzaTirociniCorsoDocente.jsp -> inserito pulsante per terminare il tirocinio. Inserita colonna "Data fine" e colonna "Stato". Inserita tabella dei tirocini conclusi contenente sia i tirocini terminati con successo, sia i tirocini scaduti. Questi ultimi vengono visualizzati sulla base del controllo tra la data odierna e la data di scadenza, pertanto non avviene una modifica all'interno del db (per semplicità si è scelta la sola modifica front-end).

ACTUAL IMPACT SET - 7 componenti impattate

DomandaTirocinio -> aggiunta del **nuovo stato** = TERMINATA.

DomandaTirocinioController -> aggiunta **1 metodo** Post terminaTirocinio. *Modifica del metodo VisualizzaTirociniInCorsoDocente in quanto è necessario reperire i tirocini in corso e terminati per visualizzarli nell'apposita area riservata.*

DomandaTirocinioService -> aggiunta **1 metodo** terminaTirocinio. All'interno di questo metodo avviene il passaggio di stato del tirocinio da APPROVATA a TERMINATA e viene modificata la data di termine del tirocinio con la data odierna.

MailSingletonSender -> modifica di **un metodo**, precisamente del **metodo per inviare la mail** al fine di prevedere messaggi che trattino i nuovi stati introdotti dall'aggiunta del docente.

visualizzaTirociniCorsoDocente.jsp -> inserito pulsante per terminare il tirocinio. Inserita colonna "Data fine" e colonna "Stato". Inserita tabella dei tirocini conclusi contenente sia i tirocini terminati con successo, sia i tirocini scaduti. Questi ultimi vengono visualizzati sulla base del controllo tra la data odierna e la data di scadenza, pertanto non avviene una modifica all'interno del db (per semplicità si è scelta la sola modifica front-end).

A seguito dell'implementazione non sono stati trovati falsi positivi, ma come si può ben notare dall'actual

impact set è stata individuata una componente non inclusa nel candidate impact set (componente evidenziata in blu precedentemente).

Summary degli insiemi evinti dall'analisi d'impatto:

CANDIDATE IMPACT SET - 6 componenti impattate

ACTUAL IMPACT SET - 7 componenti impattate

FALSE POSITIVE IMPACT SET - 0 componenti impattate

DISCOVERING POSITIVE IMPACT SET - 1 componente impattata

4.3 Metriche di Impact Analysis

A seguito delle operazioni di modifica, e avendo individuato gli insiemi di componenti relativi all'analisi d'impatto sono state calcolate ReCall, Precision, Adequacy ed Effectiveness, al fine di valutare la bontà dell'analisi d'impatto effettuata:

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

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

Inclusiveness = 0

Come nelle metriche inerenti all'impact analysis della change request n.1, notiamo sia stata trovata una componente impattata non prevista nell'actual impact set. Siccome l'AIS non è, nemmeno in questo caso, un sottoinsieme del CIS, l'inclusiveness sarà pari a 0, quindi non si ricopre una situazione ideale di adeguatezza. D'altra parte, precision e recall forniscono buoni valori siccome sono molto vicini ad 1, il che raffigura una buona situazione per una change request siccome CIS e AIS sono praticamente molto simili tra loro.

È comunque importante ricordare che la granularità degli elementi è molto fine, e che comunque in una situazione reale, difficilmente precision e recall sono effettivamente pari ad 1.

5. Testing

Si vogliono progettare delle attività di testing per verificare la presenza di errori ed, in caso siano presenti, risolverli. In questo capitolo si illustreranno le strategie di testing che si intende adottare, le funzionalità da testare e gli strumenti per condurre il testing.

5.1 Obiettivi e Criteri di Testing Plan

Per quanto riguarda il test plan e le strategie da adottare a seguito delle modifiche al progetto, si è deciso nuovamente di consultare il documento di testing del progetto iniziale. In particolare, si osserva come siano stati adoperati tre criteri di testing principali:

- Testing white-box per il testing di unità, con criterio branch coverage
- Testing Bottom-up per l'integrazione
- Testing Black-Box con criterio di category partition per il testing di sistema
- Testing di Regressione

Tra le scelte adoperate durante il progetto di sviluppo si osserva come sia stata data notevole importanza al testing riguardante la logica applicativa e la comunicazione tra essa e il database e minore priorità al package web, che contiene le interfacce controller per la comunicazione con il front-end. Tale scelta fu presa sia per ragioni di budget, ma anche in ragione a quello che è uno dei design pattern strutturali adoperato nel sistema: il Service Layer Pattern. In ragione di ciò il team di manutenzione ritiene opportuno procedere allo stesso modo per testare la correttezza degli interventi effettuati tralasciando quindi il Web Layer.

5.2 Metodologie e Livelli del testing

Quindi, il testing delle modifiche apportate sarà strutturato nel seguente modo:

Testing di unità: come obiettivo principale del test di unità si vuole verificare la correttezza dei nuovi metodi di business aggiunti e di quelli modificati a seguito degli interventi di manutenzione, con particolare riferimento al nuovo attore introdotto: il Tutor Accademico. Quindi si procederà se seguente modo:

- Aggiunta e modifica di metodi di test relativi all'iter di inserimento e approvazione della domanda di tirocinio, intervenendo opportunamente nel package di test "domanda tirocinio", secondo metodologia white-box, partendo quindi dalla classe di test domandaDITirocinioServiceUT.
- Aggiunta della classe DocenteTutorServiceUT, per il testing dei metodi di servizio relativi al nuovo utente docente tutor;

Testing di integrazione: In riferimento al testing di integrazione effettuato nel progetto iniziale, il testing di integrazione si concentrerà sul testare l'integrazione tra classi repository e database e

successivamente tra classi repository testate e classi service, quindi, si procederà nel seguente modo:

1. Creazione della classe di test DocenteTutorRepositoryIT per verificare la correttezza di interazione tra il db e la classe DocenteTutorRepository.;
2. Creazione della classe DocenteTutorServiceRepositoriesIT per verificare la correttezza di interazione tra le classi repository che sono usate nella classe DocenteTutorService;
3. Successivamente, avendo garanzia del corretto funzionamento delle componenti testate ai punti precedenti, è necessario procedere anche a ri-testare le aree di integrazione impattate dalle change request, quindi si procederà alla modifica, ri-adattamento e ri-esecuzione dei test delle classi domandaTirocinioRepositoryIT e domandaTirocinioServiceRepositoryIT.

Testing di sistema: riprendendo ciò che è stato fatto nel processo di sviluppo si è visto che è stata data molta importanza al testing funzionale delle interfacce I/O (i form), mentre le funzionalità che invocano direttamente metodi di business tramite semplice pressione di un bottone sono state ritenute per ragioni di budget trascurabili in quanto già era disponibile la garanzia di funzionamento grazie al testing di unità e di integrazione.

Considerando che le nuove funzionalità introdotte riguardano sia la presenza di un nuovo attore che accede alla piattaforma, sia il suo coinvolgimento all'interno dell'iter di inserimento di una domanda di tirocinio, si è deciso di rimodulare e ri-testare a livello di sistema (con criterio black-box category partition) le interfacce funzionali inerenti alle funzionalità login e invio domanda tirocinio.

Testing di regressione: Avendo a disposizione tutti i metodi di test anche delle componenti non impattate dal progetto di sviluppo, si è deciso di procedere con la ri-esecuzione di tutti i casi di test, per verificare la presenza di anomalie aggiuntive e il raggiungimento di una percentuale di Branch Coverage accettabile, facendo riferimento ai criteri adoperati anche nel progetto di sviluppo, il team di manutenzione ritiene opportuno ottenere un livello di Branch Coverage Superiore al 75%.

[\[Fonte IFY TPv 1.0\]](#)

Seguono i test case previsti da eseguire per il testing di sistema.

5.3 Specifiche Testing di sistema

Test Case Specification Login Piattaforma

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

Parametro: E-mail	
Formato: $\wedge \backslash w+([.]? \backslash w+)^* @ \backslash w+([.]\{1\} \backslash w+)^*([.]\{1\} \backslash w\{2,3\})\{1\} \$$	
Categorie	Scelte
Formato fe	<p>1: non rispetta il formato [errore]</p> <p>2: rispetta il formato [property formatoFEok]</p>
Esiste nel database ee	<p>1. Non esiste nel database [if formatoFEok] [errore]</p> <p>2. Esiste nel database [if formatoFEok] [property esisteEEok]</p>

Parametro: Password	
Categorie	Scelte
Corrispondenza Password cp	<p>1: La password non corrisponde [errore]</p> <p>2: La password corrisponde [property corrispondenzapasswordCPok]</p>

Codice	Combinazione	Esito
TC_LP_1	fe1	errore
TC_LP_2	fe2.ee1	errore
TC_LP_3	fe2.ee2.cp1	errore
TC_LP_4	fe2.ee2.cp2	successo

Test Case ID	TC_LP_1						
Pre-condition:							
L'ospite richiede 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:							
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Email</td><td></td></tr> <tr> <td>Password</td><td>Password1</td></tr> </table>		Input	Valore	Email		Password	Password1
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 richiede 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:							
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Email</td><td>p.verdi@unisa.it</td></tr> <tr> <td>Password</td><td>Password1</td></tr> </table>		Input	Valore	Email	p.verdi@unisa.it	Password	Password1
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.							

Test Case ID	TC_LP_3						
Pre-condition:							
L'ospite richiede 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:							
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Email</td><td>m.rossi@unisa.it</td></tr> <tr> <td>Password</td><td>Password2</td></tr> </table>		Input	Valore	Email	m.rossi@unisa.it	Password	Password2
Input	Valore						
Email	m.rossi@unisa.it						
Password	Password2						
2. L'ospite clicca sul pulsante "Accedi"							
Oracle:							
Il sistema visualizza il messaggio: "Password non valida". Il login non viene effettuato.							

Test Case ID	TC_LP_4						
Pre-condition:							
L'ospite richiede 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:							
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Email</td><td>m.rossi@unisa.it</td></tr> <tr> <td>Password</td><td>Password1</td></tr> </table>		Input	Valore	Email	m.rossi@unisa.it	Password	Password1
Input	Valore						
Email	m.rossi@unisa.it						
Password	Password1						
2. L'ospite clicca sul pulsante "Accedi"							
Oracle:							
Il login viene effettuato.							

Test Case Specification Invio domanda di tirocinio

Parametro	Tipo	Numero caratteri
Conoscenze Tecniche	String	200
Motivazioni	String	300
Id Docente Tutor	Integer	N/A
Data Inizio Tirocinio	Date	N/A
Data Fine Tirocinio	Date	N/A
Numero CFU Associati	Integer	N/A
Conferma regolamento sulla privacy	Boolean	N/A

Parametro: Conoscenze Tecniche	
Categorie	Scelte
Lunghezza lct	1: lunghezza < 1 [errore] 2: lunghezza > 200 [errore] 3: lunghezza >= 1 && lunghezza <= 200 [property lunghezzaLCTok]

Parametro: Motivazioni	
Categorie	Scelte
Lunghezza lm	<p>1: lunghezza < 1 [errore]</p> <p>2: lunghezza > 300 [errore]</p> <p>3: lunghezza >= 1 && lunghezza <= 300 [property lunghezzaLMok]</p>

Parametro: ID docente Tutor	
Categorie	Scelte
Valorizzazione Campo Docente VCD	<p>1: campo vuoto [errore]</p> <p>2: campo non vuoto [campoVCDok]</p>
Esiste nel database EDD	<p>1: non esiste nel database [if campoVCDok] [errore]</p> <p>2: esiste nel database [campoEDDok]</p>

Parametro: Data Inizio Tirocinio	
Categorie	Scelte
datainizio di	<p>1: datainizio < datacorrente [errore]</p> <p>2: datainizio > datacorrente [property datainizioDlok]</p>

Parametro: Data Fine Tirocinio	
Categorie	Scelte
datafine df	1: datafine < datainizio [errore] 2: datafine > datainizio [property datafineDFok]

Parametro: Numero CFU Associati	
Categorie	Scelte
valore vcfu	1: valore<6 valore>12 [errore] 2: 6<=valore<=12 [property valoreVCFUok]

Parametro: Conferma regolamento sulla privacy	
Formato: ^Accetto\$	
Categorie	Scelte
Regolamento r	1: checkbox non spuntato [errore] 2: checkbox spuntato [property regolamentoRok]

Codice	Combinazione	Esito
TC_IDT_1	lct1	errore
TC_IDT_2	lct2	errore
TC_IDT_3	lct3.lm1	errore
TC_IDT_4	lct3.lm2	errore
TC_IDT_5	lct3.lm3.vcd1	errore
TC_IDT_6	lct3.lm3.vcd2.edd1	errore
TC_IDT_7	lct3.lm3.vcd2.edd2.di1	errore
TC_IDT_8	lct3.lm3.vcd2.edd2.di2.df1	errore
TC_IDT_9	lct3.lm3.vcd2.edd2.di2.df2.vcfu1	errore
TC_IDT_10	lct3.lm3.vcd2.edd2.di2.df2.vcfu2.r1	errore
TC_IDT_11	lct3.lm3.vcd2.edd2.di2.df2.vcfu2.r2	successo

Test Case ID	TC_IDT_1
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "il campo Conoscenze Tecniche deve contenere almeno un carattere". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_2
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "il campo Conoscenze Tecniche deve contenere al massimo 200 caratteri.". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_3																
Pre-condition:																	
Lo studente richiede l'invio di una domanda di tirocinio.																	
Flow of events:																	
1. Lo studente compila i campi richiesti:																	
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Conoscenze Tecniche</td><td>Buona conoscenza di Java</td></tr> <tr> <td>Motivazioni</td><td></td></tr> <tr> <td>ID del docente</td><td>4</td></tr> <tr> <td>Data inizio tirocinio</td><td>27/07/2021</td></tr> <tr> <td>Data fine tirocinio</td><td>16/12/2021</td></tr> <tr> <td>Numero CFU associati</td><td>6</td></tr> <tr> <td>Conferma regolamento privacy</td><td>Checkbox spuntato</td></tr> </table>		Input	Valore	Conoscenze Tecniche	Buona conoscenza di Java	Motivazioni		ID del docente	4	Data inizio tirocinio	27/07/2021	Data fine tirocinio	16/12/2021	Numero CFU associati	6	Conferma regolamento privacy	Checkbox spuntato
Input	Valore																
Conoscenze Tecniche	Buona conoscenza di Java																
Motivazioni																	
ID del docente	4																
Data inizio tirocinio	27/07/2021																
Data fine tirocinio	16/12/2021																
Numero CFU associati	6																
Conferma regolamento privacy	Checkbox spuntato																
2. L'ospite clicca sul pulsante "Invia Domanda".																	
Oracle:																	
Il sistema visualizza il messaggio "il campo Motivazioni deve contenere almeno un carattere". La domanda di tirocinio non viene inviata.																	

Test Case ID	TC_IDT_4
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaa
ID del docente	4
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	

Il sistema visualizza il messaggio "il campo Motivazioni deve contenere al massimo 300 caratteri.". La domanda di tirocinio non viene inviata.

Test Case ID	TC_IDT_5
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "Non è stato scelto alcun docente". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_6
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	20
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "il docente immesso non è valido". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_7
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/12/2020
Data fine tirocinio	16/12/2026
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "il campo Data Inizio Tirocinio deve contenere una data maggiore della data corrente". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_8
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/12/2025
Data fine tirocinio	16/12/2022
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "il campo Data Fine Tirocinio deve contenere una data di fine maggiore della data di inizio del tirocinio". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_9																
Pre-condition:																	
Lo studente richiede l'invio di una domanda di tirocinio.																	
Flow of events:																	
1. Lo studente compila i campi richiesti:																	
<table> <tr> <th>Input</th><th>Valore</th></tr> <tr> <td>Conoscenze Tecniche</td><td>Buona conoscenza di Java</td></tr> <tr> <td>Motivazioni</td><td>Arricchimento delle conoscenze</td></tr> <tr> <td>ID del docente</td><td>4</td></tr> <tr> <td>Data inizio tirocinio</td><td>27/07/2021</td></tr> <tr> <td>Data fine tirocinio</td><td>16/12/2021</td></tr> <tr> <td>Numero CFU associati</td><td>24</td></tr> <tr> <td>Conferma regolamento privacy</td><td>Checkbox spuntato</td></tr> </table>		Input	Valore	Conoscenze Tecniche	Buona conoscenza di Java	Motivazioni	Arricchimento delle conoscenze	ID del docente	4	Data inizio tirocinio	27/07/2021	Data fine tirocinio	16/12/2021	Numero CFU associati	24	Conferma regolamento privacy	Checkbox spuntato
Input	Valore																
Conoscenze Tecniche	Buona conoscenza di Java																
Motivazioni	Arricchimento delle conoscenze																
ID del docente	4																
Data inizio tirocinio	27/07/2021																
Data fine tirocinio	16/12/2021																
Numero CFU associati	24																
Conferma regolamento privacy	Checkbox spuntato																
2. L'ospite clicca sul pulsante "Invia Domanda".																	
Oracle:																	
Il sistema visualizza il messaggio "il campo Numero CFU Associati deve contenere valori che vadano da 6 a 12". La domanda di tirocinio non viene inviata.																	

Test Case ID	TC_IDT_10
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox non spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "è obbligatorio accettare le condizioni sulla privacy". La domanda di tirocinio non viene inviata.	

Test Case ID	TC_IDT_11
Pre-condition:	
Lo studente richiede l'invio di una domanda di tirocinio.	
Flow of events:	
1. Lo studente compila i campi richiesti:	
Input	Valore
Conoscenze Tecniche	Buona conoscenza di Java
Motivazioni	Arricchimento delle conoscenze
ID del docente	4
Data inizio tirocinio	27/07/2021
Data fine tirocinio	16/12/2021
Numero CFU associati	6
Conferma regolamento privacy	Checkbox spuntato
2. L'ospite clicca sul pulsante "Invia Domanda".	
Oracle:	
Il sistema visualizza il messaggio "La sua domanda di tirocinio è stata inserita con successo.". La domanda di tirocinio viene inviata.	

5.4 Testing Excution Report

Testing di Unità

A seguito del testing di unità WhiteBox delle componenti impattate dalla modifica rispettivamente nei package DomandaTirocinio e nel nuovo package per il docente tutor si ottengono i seguenti risultati di Coverage:

DomandaTirocinioServiceUT

Coverage: DomandaTirocinioServiceUT ×				
100% classes, 90% lines covered in package 'it.unisa.di.is.gc1.ify.domandaTirocinio'				
Element	Class, %	Method, %	Line, %	
DomandaTirocinio	100% (1/1)	88% (22/25)	54% (33/61)	
DomandaTirocinioNonValidaException	100% (1/1)	80% (4/5)	80% (8/10)	
DomandaTirocinioService	100% (1/1)	100% (28/28)	98% (275/279)	

DocenteTutorServiceUT

Coverage: it.unisa.di.is.gc1.ify in ify ×				
100% classes, 68% lines covered in package 'it.unisa.di.is.gc1.ify.DocenteTutor'				
Element	Class, %	Method, %	Line, %	Branch, %
DocenteTutor	100% (1/1)	60% (3/5)	44% (4/9)	100% (0/0)
DocenteTutorService	100% (1/1)	100% (2/2)	100% (7/7)	100% (0/0)

Testing di Integrazione

A seguito del testing di integrazione Bottom-UP tra i layer DB - Repositories e Repositories-Services impattati dalla modifica, si ottengono i seguenti risultati di Coverage:

DocenteTutorRepositoryIT

Coverage: DomandaTirocinioRepositoryIT x DocenteTutorRepositoryIT x

50% classes, 20% lines covered in package 'it.unisa.di.is.gc1.ify.DocenteTutor'

Element	Class, %	Method, %	Line, %	Branch, %
DocenteTutor	100% (1/1)	40% (2/5)	33% (3/9)	100% (0/0)
DocenteTutorService	0% (0/1)	0% (0/2)	0% (0/6)	100% (0/0)

DocenteTutorServiceRepositoriesIT;

Coverage: DocenteTutorServiceRepositoryIT x

100% classes, 62% lines covered in package 'it.unisa.di.is.gc1.ify.DocenteTutor'

Element	Class, %	Method, %	Line, %	Branch, %
DocenteTutor	100% (1/1)	40% (2/5)	33% (3/9)	100% (0/0)
DocenteTutorService	100% (1/1)	100% (2/2)	100% (7/7)	100% (0/0)

DomandaTirocinioRepositoryIT

Coverage: DomandaTirocinioRepositoryIT x

33% classes, 7% lines covered in package 'it.unisa.di.is.gc1.ify.domandaTirocinio'

Element	Class, %	Method, %	Line, %	Branch, %
DomandaTirocinio	100% (1/1)	60% (15/25)	40% (25/61)	100% (0/0)
DomandaTirocinioNonValidaException	0% (0/1)	0% (0/5)	0% (0/10)	100% (0/0)
DomandaTirocinioService	0% (0/1)	0% (0/28)	0% (0/278)	0% (0/166)

DomandaTirocinioServiceRepositoryIT.

Coverage: DomandaTirocinioServiceRepositoriesIT x

66% classes, 65% lines covered in package 'it.unisa.di.is.gc1.ify.domandaTirocinio'

Element	Class, %	Method, %	Line, %	Branch, %
DomandaTirocinio	100% (1/1)	84% (21/25)	73% (45/61)	100% (0/0)
DomandaTirocinioNonValidaException	0% (0/1)	0% (0/5)	0% (0/10)	100% (0/0)
DomandaTirocinioService	100% (1/1)	71% (20/28)	66% (185/279)	38% (64/166)

Testing di sistema

Testing per Funzionalità di Login Utente

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

TC_LP_1 Execution Screen

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

TC_LP_2 Execution Screen

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

TC_LP_3 Execution Screen

Running 'TC_LP_3'

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 m.rossi@unisa.it OK
6. click on id=inputPassword OK
7. type on id=inputPassword with value Password2 OK
8. click on id=reg OK
9. assertText on css=.myError with value Password non valida OK

'TC_LP_3' completed successfully

TC_LP_4 Execution Screen

Running 'TC_LP_4'

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 m.rossi@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. assertElementPresent on css=.my-4 OK

'TC_LP_4' completed successfully

Testing per Funzionalità di Invio domanda tirocinio

Test Case	Data e Ora Esecuzione	Pass/Fail
TC_IDT_1	16-07-2021 / 17:20	Pass
TC_IDT_2	16-07-2021 / 17:20	Pass
TC_IDT_3	16-07-2021 / 17:20	Pass
TC_IDT_4	16-07-2021 / 17:20	Pass
TC_IDT_5	16-07-2021 / 17:20	Pass
TC_IDT_6	16-07-2021 / 17:20	Pass
TC_IDT_7	16-07-2021 / 17:20	Pass
TC_IDT_8	16-07-2021 / 17:20	Pass
TC_IDT_9	16-07-2021 / 17:20	Pass
TC_IDT_10	16-07-2021 / 17:20	Pass
TC_IDT_11	16-07-2021 / 17:20	Pass

TC_IDT_1 Execution Screen

Running 'TC_IDT_1'

1. open on http://localhost:8080/ OK
 2. setWindowSize on 1413x904 OK
 3. click on linkText=Aziende convenzionate OK
 4. click on linkText=Progetti Formativi OK
 5. click on css=.detail-view td:nth-child(6) .btn OK
 6. click on id=formGroupExampleInput OK
 7. click on id=docenteTutorId OK
 8. click on id=docenteTutorId OK
 9. click on name=dataInizio OK
 10. type on name=dataInizio with value 2021-07-27 OK
 11. click on name=dataFine OK
 12. type on name=dataFine with value 2021-12-16 OK
 13. click on name=numeroCFU OK
 14. type on name=numeroCFU with value 6 OK
 15. click on id=formGroupExampleInput2 OK
 16. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 17. click on id=accetto OK
 18. click on css=.btn:nth-child(10) OK
 19. assertText on css=.form-group:nth-child(2) > .myError with value il campo Conoscenze Tecniche deve contenere almeno un carattere. OK
- 'TC_IDT_1' completed successfully**

Running 'TC_IDT_2'

- open on http://localhost:8080/ OK
 - setWindowSize on 1413x904 OK
 - click on linkText=Aziende convenzionate OK
 - click on linkText=Progetti Formativi OK
 - click on css=.detail-view td:nth-child(6) .btn OK
 - click on id=formGroupExampleInput OK
 - type on id=formGroupExampleInput with value
aa
 - click on id=docenteTutorId OK
-
- click on id=docenteTutorId OK
 - click on name=dataInizio OK
 - type on name=dataInizio with value 2021-07-27 OK
 - click on name=dataFine OK
 - type on name=dataFine with value 2021-12-16 OK
 - click on name=numeroCFU OK
 - type on name=numeroCFU with value 6 OK
 - click on id=formGroupExampleInput2 OK
 - type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 - click on id=accetto OK
 - click on css=.btn:nth-child(10) OK
 - assertText on css=.form-group:nth-child(2) > .myError with value il campo Conoscenze Tecniche deve contenere al massimo 200 caratteri. OK
- 'TC_IDT_2' completed successfully

TC_IDT_3 Execution Screen

Running 'TC_IDT_3'

1. open on http://localhost:8080/visualizzaAziendeConvenzionateStudiante OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Progetti Formativi OK
4. click on css=.detail-view td:nth-child(6) .btn OK
5. click on id=formGroupExampleInput OK
6. click on id=formGroupExampleInput OK
7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
8. click on id=docenteTutorId OK
9. click on id=docenteTutorId OK
10. click on id=formGroupExampleInput2 OK
11. click on css=.form-group:nth-child(4) OK
12. click on name=numeroCFU OK
13. type on name=numeroCFU with value 6 OK
14. click on name=dataInizio OK
15. click on name=dataInizio OK
16. type on name=dataInizio with value 0002-07-27 OK
17. type on name=dataInizio with value 0020-07-27 OK
18. type on name=dataInizio with value 0202-07-27 OK
19. type on name=dataInizio with value 2021-07-27 OK
20. click on name=dataFine OK
21. type on name=dataFine with value 0002-12-16 OK
22. type on name=dataFine with value 0020-12-16 OK
23. type on name=dataFine with value 0202-12-16 OK
24. type on name=dataFine with value 2021-12-16 OK
25. click on css=.form-check OK
26. click on id=accetto OK
27. click on css=.btn:nth-child(10) OK
28. assertText on css=.myError with value il campo Motivazioni deve contenere almeno un carattere. OK

'TC_IDT_3' completed successfully

TC_IDT_4 Execution Screen

Running 'TC_IDT_4'

1. open on http://localhost:8080/visualizzaAziendeConvenzionateStudente OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Progetti Formativi OK
4. click on css=.detail-view td:nth-child(6) .btn OK
5. click on id=formGroupExampleInput OK
6. click on id=formGroupExampleInput OK
7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
8. click on id=docenteTutorId OK
9. click on id=docenteTutorId OK
10. click on id=formGroupExampleInput2 OK

type on id=formGroupExampleInput2 with value
11.aa
OK

12. click on css=.form-group:nth-child(4) OK
13. click on name=numeroCFU OK
14. type on name=numeroCFU with value 6 OK
15. click on name=dataInizio OK
16. click on name=dataInizio OK
17. type on name=dataInizio with value 0002-07-27 OK
18. type on name=dataInizio with value 0020-07-27 OK
19. type on name=dataInizio with value 0202-07-27 OK
20. type on name=dataInizio with value 2021-07-27 OK
21. click on name=dataFine OK

22. type on name=dataFine with value 0002-12-16 OK

23. type on name=dataFine with value 0020-12-16 OK

24. type on name=dataFine with value 0202-12-16 OK

25. type on name=dataFine with value 2021-12-16 OK

26. click on css=.form-check OK

27. click on id=accetto OK

28. click on css=.btn:nth-child(10) OK

29. assertText on css=.myError with value il campo Motivazioni deve contenere al massimo 300 caratteri. OK

'TC_IDT_4' completed successfully

TC_IDT_5 Execution Screen

Running 'TC_IDT_5'

1. open on http://localhost:8080/visualizzaAziendeConvenzionateStudiante OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Progetti Formativi OK
4. click on css=.detail-view td:nth-child(6) .btn OK
5. click on id=formGroupExampleInput OK
6. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
7. click on id=docenteTutorId OK
8. click on id=formGroupExampleInput2 OK
9. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
10. click on name=numeroCFU OK
11. type on name=numeroCFU with value 6 OK
12. click on css=.modal-body > form OK
13. click on name=datInizio OK
14. type on name=datInizio with value 0002-07-27 OK
15. type on name=datInizio with value 0020-07-27 OK
16. type on name=datInizio with value 0202-07-27 OK
17. type on name=datInizio with value 2021-07-27 OK
18. click on name=dataFine OK
19. click on name=dataFine OK
20. type on name=dataFine with value 0002-12-16 OK
21. type on name=dataFine with value 0020-12-16 OK
22. type on name=dataFine with value 0202-12-16 OK
23. type on name=dataFine with value 2021-12-16 OK
24. click on id=accetto OK
25. click on css=.btn:nth-child(10) OK
26. assertText on css=.myError with value Non è stato scelto alcun docente OK

'TC_IDT_5' completed successfully

TC_IDT_6 Execution Screen

Running 'TC_IDT_6'

1. open on http://localhost:8080/visualizzaAziendeConvenzionateStudente OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Progetti Formativi OK
4. click on css=.detail-view td:nth-child(6) .btn OK
5. click on id=formGroupExampleInput OK
6. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
7. click on id=docenteTutorId OK
8. click on id=formGroupExampleInput2 OK
9. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
10. click on name=numeroCFU OK
11. type on name=numeroCFU with value 6 OK
12. click on css=.modal-body > form OK
13. click on name=dataInizio OK
14. type on name=dataInizio with value 0002-07-27 OK
15. type on name=dataInizio with value 0020-07-27 OK
16. type on name=dataInizio with value 0202-07-27 OK
17. type on name=dataInizio with value 2021-07-27 OK
18. click on name=dataFine OK
19. click on name=dataFine OK
20. type on name=dataFine with value 0002-12-16 OK
21. type on name=dataFine with value 0020-12-16 OK
22. type on name=dataFine with value 0202-12-16 OK
23. type on name=dataFine with value 2021-12-16 OK
24. click on id=accetto OK
25. click on css=.btn:nth-child(10) OK
26. assertText on css=.myError with value Il docente immesso non è valido OK

'TC_IDT_6' completed successfully

TC_IDT_7 Execution Screen

Running 'TC_IDT_7'

1. open on http://localhost:8080/ OK
 2. setWindowSize on 1552x840 OK
 3. click on linkText=Aziende convenzionate OK
 4. click on linkText=Progetti Formativi OK
 5. click on css=.detail-view td:nth-child(6) .btn OK
 6. click on id=formGroupExampleInput OK
 7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
 8. click on id=docenteTutorId OK
 9. click on id=docenteTutorId OK
 10. click on id=formGroupExampleInput2 OK
 11. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 12. click on name=numeroCFU OK
 13. type on name=numeroCFU with value 6 OK
 14. click on name=dataInizio OK
 15. type on name=dataInizio with value 0002-12-27 OK
 16. type on name=dataInizio with value 0020-12-27 OK
 17. type on name=dataInizio with value 0202-12-27 OK
 18. type on name=dataInizio with value 2020-12-27 OK
 19. click on name=dataFine OK
 20. type on name=dataFine with value 0002-12-16 OK
 21. type on name=dataFine with value 0020-12-16 OK
 22. type on name=dataFine with value 0206-12-16 OK
 23. type on name=dataFine OK
 24. type on name=dataFine with value 0002-12-16 OK
 25. type on name=dataFine with value 0020-12-16 OK
 26. type on name=dataFine with value 0202-12-16 OK
 27. type on name=dataFine with value 2026-12-16 OK
 28. click on id=accetto OK
 29. click on css=.btn:nth-child(10) OK
 30. assertText on css=.myError with value Il campo Data Inizio Tirocinio deve contenere una data maggiore della data corrente. OK
- 'TC_IDT_7' completed successfully**

TC_IDT_8 Execution Screen

Running 'TC_IDT_8'

1. open on http://localhost:8080/ OK
 2. setWindowSize on 1552x840 OK
 3. click on linkText=Aziende convenzionate OK
 4. click on linkText=Progetti Formativi OK
 5. click on css=.detail-view td:nth-child(6) .btn OK
 6. click on id=formGroupExampleInput OK
 7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
 8. click on id=docenteTutorId OK
 9. click on id=docenteTutorId OK
 10. click on id=formGroupExampleInput2 OK
 11. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 12. click on name=numeroCFU OK
 13. type on name=numeroCFU with value 6 OK
 14. click on name=dataInizio OK
 15. type on name=dataInizio with value 0002-12-16 OK
 16. type on name=dataInizio with value 0020-12-16 OK
 17. type on name=dataInizio with value 0202-12-16 OK
 18. type on name=dataInizio with value 2022-12-16 OK
 19. click on name=dataInizio OK
 20. click on name=dataInizio OK
 21. doubleClick on name=dataInizio OK
 22. type on name=dataInizio with value 2022-12-02 OK
 23. type on name=dataInizio with value 2022-12-27 OK
 24. click on name=dataInizio OK
 25. type on name=dataInizio with value 0002-12-27 OK
 26. type on name=dataInizio with value 0020-12-27 OK
 27. type on name=dataInizio with value 0205-12-27 OK
 28. type on name=dataInizio OK
 29. type on name=dataInizio with value 0002-12-27 OK
 30. type on name=dataInizio with value 0020-12-27 OK
 31. type on name=dataInizio with value 0202-12-27 OK
 32. type on name=dataInizio with value 2025-12-27 OK
 33. click on name=dataFine OK
 34. type on name=dataFine with value 0002-12-16 OK
 35. type on name=dataFine with value 0020-12-16 OK
 36. type on name=dataFine with value 0202-12-16 OK
 37. type on name=dataFine with value 2020-12-16 OK
 38. type on name=dataFine OK
 39. type on name=dataFine with value 0002-12-16 OK
 40. type on name=dataFine with value 0020-12-16 OK
 41. type on name=dataFine with value 0202-12-16 OK
 42. type on name=dataFine with value 2022-12-16 OK
 43. click on id=accetto OK
 44. click on css=.btn:nth-child(10) OK
 45. assertText on css=.myError with value il campo Data Fine Tirocinio deve contenere una data di fine maggiore della data di inizio del tirocinio. OK
- 'TC_IDT_8' completed successfully

TC_IDT_9 Execution Screen

Running 'TC_IDT_9'

1. open on http://localhost:8080/ OK
 2. setWindowSize on 1552x840 OK
 3. click on linkText=Aziende convenzionate OK
 4. click on linkText=Progetti Formativi OK
 5. click on css=.detail-view td:nth-child(6) .btn OK
 6. click on id=formGroupExampleInput OK
 7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
 8. click on id=docenteTutorId OK
 9. click on id=docenteTutorId OK
 10. click on id=formGroupExampleInput2 OK
 11. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 12. click on name=numeroCFU OK
 13. type on name=numeroCFU with value 24 OK
 14. click on name=dataInizio OK
 15. type on name=dataInizio with value 0002-07-27 OK
 16. type on name=dataInizio with value 0020-07-27 OK
 17. type on name=dataInizio with value 0202-07-27 OK
 18. type on name=dataInizio with value 2021-07-27 OK
 19. click on name=dataFine OK
 20. type on name=dataFine with value 0002-12-16 OK
 21. type on name=dataFine with value 0020-12-16 OK
 22. type on name=dataFine with value 0202-12-16 OK
 23. type on name=dataFine with value 2021-12-16 OK
 24. click on id=accetto OK
 25. click on css=.btn:nth-child(10) OK
26. assertText on css=.myError with value Il campo Numero CFU Associati deve contenere valori che vadano da 6 a 12. OK
- 'TC_IDT_9' completed successfully**
-

TC_IDT_10 Execution Screen

Running 'TC_IDT_10'

1. open on http://localhost:8080/ OK
2. setWindowSize on 1552x840 OK
3. click on linkText=Aziende convenzionate OK
4. click on linkText=Progetti Formativi OK
5. click on css=.detail-view td:nth-child(6) .btn OK
6. click on id=formGroupExampleInput OK
7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
8. click on id=docenteTutorId OK
9. click on id=docenteTutorId OK
10. click on id=formGroupExampleInput2 OK
11. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
12. click on name=numeroCFU OK
13. type on name=numeroCFU with value 6 OK
14. click on name=dataInizio OK
15. type on name=dataInizio with value 0002-07-27 OK
16. type on name=dataInizio with value 0020-07-27 OK
17. type on name=dataInizio with value 0202-07-27 OK
18. type on name=dataInizio with value 2021-07-27 OK
19. click on name=dataFine OK
20. type on name=dataFine with value 0002-12-16 OK
21. type on name=dataFine with value 0020-12-16 OK
22. type on name=dataFine with value 0202-12-16 OK
23. type on name=dataFine with value 2021-12-16 OK
24. click on css=.btn:nth-child(10) OK
25. assertText on css=.myError with value È obbligatorio accettare le condizioni sulla privacy. OK

'TC_IDT_10' completed successfully

TC_IDT_11 Execution Screen

Running 'TC_IDT_11'

1. open on http://localhost:8080/ OK
 2. setWindowSize on 1296x896 OK
 3. click on linkText=Aziende convenzionate OK
 4. click on linkText=Progetti Formativi OK
 5. click on css=.detail-view td:nth-child(6) .btn OK
 6. click on id=formGroupExampleInput OK
 7. type on id=formGroupExampleInput with value Buona conoscenza di Java OK
 8. click on id=docenteTutorId OK
 9. click on id=docenteTutorId OK
 10. click on id=formGroupExampleInput2 OK
 11. type on id=formGroupExampleInput2 with value Arricchimento delle conoscenze OK
 12. click on name=numeroCFU OK
 13. type on name=numeroCFU with value 6 OK
 14. click on name=dataInizio OK
 15. type on name=dataInizio with value 0002-07-27 OK
 16. type on name=dataInizio with value 0020-07-27 OK
 17. type on name=dataInizio with value 0202-07-27 OK
 18. type on name=dataInizio with value 2021-07-27 OK
 19. click on name=dataFine OK
 20. type on name=dataFine with value 0002-12-16 OK
 - ...
 21. type on name=dataFine with value 0020-12-16 OK
 22. type on name=dataFine with value 0202-12-16 OK
 23. type on name=dataFine with value 2021-12-16 OK
 24. click on id=accetto OK
 25. click on css=.btn:nth-child(10) OK
 26. assertText on css=strong with value La sua domanda di tirocinio è stata inserita con successo. OK
- 'TC_IDT_11' completed successfully**

Testing di regressione

Come anticipato nelle metodologie adoperate, è stata applicata la metodologia “Retest all” con tutte le classi di test (di unità e integrazione) disponibili al termine della fase di testing del progetto di manutenzione. I risultati finali ottenuti sono i seguenti:

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)

Come si osserva dal report finale, si ottiene una Branch Coverage dell'85%, ben oltre il limite del 75% prefissato, ad influire molto in termini percentuali, ovviamente è la scelta di non procedere con il testing del layer di presentazione del Package “WEB” scelta derivante dal progetto di sviluppo, adoperata anche per l'attuale progetto di manutenzione come già anticipato e motivato. D'altro canto, infatti, se si osserva però la statistica di dettaglio dei singoli package, si nota come quasi il 100% dei Branch delle componenti coinvolte dalle test suite a disposizione del team vengano coperti.