



TirocinioSmart

IAD – Impact Analysis Document

TirocinioSmart

Data	Versione	Cambiamenti	Autori
10/08/2020	0.1	Stesura documento	[RC]
30/08/2020	1.0	Revisione finale	[RC]

Sommario

1. Introduzione	5
2. Scopo del documento.....	5
2.1 Definizioni, acronimi e abbreviazioni	6
3. Panoramica del sistema esistente	6
4. Analisi delle modifiche proposte	7
4.1 Soluzione individuata.....	8
4.2 Diagramma delle attività.....	9
4.2.1 Invio di una domanda di tirocinio	10
4.2.2 Aggiunta nuova attività di tirocinio	11
4.3 Requisiti funzionali.....	12
4.3.1 Servizi per la gestione delle convenzioni	13
4.3.2 Requisiti per la gestione dei progetti formativi.....	13
4.3.3 Requisiti per la gestione delle domande di tirocinio	13
4.3.4 Requisiti per la gestione dei tirocini.....	14
4.3.5 Servizi per la gestione dei registri di tirocinio	15
4.3.6 Requisiti per la gestione dell'utenza	16
4.4 Modello di sistema.....	16
4.5 Modello dinamico.....	29
4.5.1 Sequence Diagram	29
4.5.1.1 SD_GRT_2:AggiuntaAttivitàTirocinio.....	29
4.5.1.2 SD_GRT_4:ValidazioneAttivitàTirocinio	30
4.5.1.3 SD_GRT_5:AnnullamentoAttivitàTirocinio	30
4.5.1.4 SD_GRT_6:ChiusuraRegistroTirocinio.....	31
4.5.1.5 SD_GRT_8:ValidazioneRegistroTirocinio	31
4.5.1.6 SD_GDT_11:ValidazioneDomandaTirocinio.....	32
4.5.1.7 SD_GDT_12:AnnullamentoDomandaTirocinio.....	32
4.5.2 Statechart.....	32
4.5.2.1 SCD_GDT:DomandaTirocinio	33
4.5.2.2 SCD_GRT:AttivitàTirocinio.....	33
4.5.2.3 SCD_GRT:RegistroTirocinio	33
4.6 Gestione dei dati persistenti.....	33
4.7 Decomposizione in sottosistemi.....	34
5. Impact Analysis Process	36
5.1 Individuazione dello Starting Impact Set (SIS).....	37
5.1.1 01-CR1:Aggiunta Attore.....	37

5.1.2 03-CR3:Servizio di Mailing.....	38
5.2 Individuazione del Candidate Impact Set (CIS)	38
5.2.1 01-CR1:Aggiunta Attore.....	39
5.2.2 03-CR3:Servizio di Mailing.....	39
5.2.3 Candidate Set finale	40
5.3 Definizione Actual Impact Set	40
5.3.1 01-CR1:Aggiunta Attore.....	40
5.3.2 02-CR2:Servizio di Mailing.....	48
5.3.3 Actual Impact Set finale.....	48
5.4 False Positive Impact Set	49
5.5 Metriche per la valutazione del processo di Impact Analysis.....	49
6. Conclusioni	50

1. Introduzione

Un'esperienza di tirocinio è buona se permette al tirocinante di maturare un bagaglio di esperienze, competenze, realmente utili al suo ingresso nel mercato del lavoro. Tutto questo è possibile anche attraverso il supporto di TirocinioSmart. Agevolare quelle che sono le richieste degli studenti e rendere migliori le loro esperienze formative è proprio uno degli obiettivi di TirocinioSmart.

L'idea di far evolvere il sistema nasce anche con questo obiettivo: migliorare la qualità dei servizi offerti per favorire una migliore interazione tra il personale dell'ufficio tirocini, l'azienda convenzionata e lo studente che richiede di poter effettuare il tirocinio, innescando un meccanismo che semplifica il lavoro di logistica nella gestione dei ruoli.

Il cuore delle modifiche proposte si basa essenzialmente su alcune particolari esigenze:

- rendere elettronica la gestione del Registro di Tirocinio da parte di uno studente eliminando così del tutto la necessità di consegna di documenti in formato cartaceo;
- effettuare un'ulteriore validazione delle Domande di Tirocinio da parte del Presidente Del Consiglio Didattico, proprio come previsto dal regolamento ufficiale, e al fine di poter garantire una gestione più sicura ed efficace di tutte le domande che giungono al Dipartimento di Informatica dell'Università Degli Studi Di Salerno;
- realizzare un servizio di mailing avente lo scopo di informare l'utente riguardo al verificarsi di determinati eventi.

Per ulteriori dettagli relativamente alle modifiche proposte, si rimanda al documento di specifica delle change request.

2. Scopo del documento

Il documento di Impact Analysis si propone di stabilire quali parti della struttura del sistema saranno influenzate dalle modifiche, che sono state specificate nel documento di CR, tramite un attento esame del codice sorgente.

I vantaggi che possiamo trarre dalla realizzazione di questo documento sono molteplici:

- individuando i componenti che devono essere modificati per ospitare i miglioramenti proposti si possono velocizzare considerevolmente gli interventi di modifica (con una conseguente riduzione dei costi);
- la tracciabilità mantenuta nella fase di realizzazione di TirocinioSmart tra i requisiti del sistema ed i moduli che li implementano, consente di verificare la conformità delle modifiche che si vogliono introdurre rispetto alle specifiche del progetto originario;
- determinando gli effetti “ripple” causati dalla modifica in questione, si riducono gli interventi di manutenzione correttiva dovuti ad errori introdotti dalle nuove modifiche.

L'Impact Analysis si configura quindi come un metodo che permette di effettuare una manutenzione efficiente e al contempo estremamente efficace, poiché consente di scegliere il tipo di intervento più adeguato per il sistema in esame.

2.1 Definizioni, acronimi e abbreviazioni

Nel corso del documento si farà riferimento alla seguente distinzione:

- *“accettazione”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dal delegato aziendale che intende manifestare interesse nell'accogliere all'interno della propria azienda lo studente che ha presentato la domanda. Accettare una domanda consiste sostanzialmente nell'inoltrare la stessa all'Ufficio Tirocini per un'ulteriore approvazione.
- *“rifiuto”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dal delegato aziendale che non ritiene lo studente adeguato al progetto formativo selezionato oppure al profilo dell'azienda che rappresenta.
- *“approvazione”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dall'impiegato dell'Ufficio Tirocini quando intende confermare l'attivazione di un tirocinio (a partire da una domanda accettata dall'azienda) dopo averne esaminato la domanda. Approvare una domanda di tirocinio consiste sostanzialmente nell'inoltrare la stessa al Presidente Del Consiglio Didattico per la validazione finale.
- *“respinta”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dall'impiegato dell'Ufficio Tirocini quando il profilo di uno studente non è ritenuto valido, bloccando così l'attivazione di un tirocinio sebbene l'azienda coinvolta abbia accettato la domanda ad egli associata.
- *“validazione”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dal Presidente Del Consiglio Didattico che intende validare la domanda in questione e procedere con l'attivazione vera e propria del Tirocinio presso la struttura ospitante.
- *“annullare”*, in riferimento ad una domanda di tirocinio, indicherà l'azione intrapresa dal Presidente Del Consiglio Didattico quando la domanda in questione non è ritenuta valida, pertanto procede all'annullamento dell'attivazione del Tirocinio.

3. Panoramica del sistema esistente

TirocinioSmart è un'applicazione che mira all'ottimizzazione e l'automatizzazione dei tirocini presso le aziende convenzionate con l'Università Degli Studi di Salerno. Con una digitalizzazione delle pratiche del tirocinio è possibile gestire tutti i passi necessari in maniera rapida e fluida.

Il sistema consiste di una piattaforma web alla quale possono avere accesso le principali entità coinvolte nella gestione dei tirocini, fornendo loro le funzionalità di cui necessitano per la corretta gestione dell'iter.

Lo **studente** può creare un account per poter interagire con il sistema: è richiesto che questo venga validato da un incaricato dell'ufficio addetto al controllo dei dati inseriti (data l'impossibilità di comunicare con il sistema esse3 su cui fa affidamento l'Università).

Successivamente alla registrazione, lo studente potrà dunque consultare l'elenco delle aziende convenzionate ed i progetti formativi da esse offerti al fine di determinare quale percorso soddisfi maggiormente le sue aspettative. Basteranno soltanto pochi click per effettuare una domanda di tirocinio: selezionato il progetto formativo (e quindi l'azienda presso cui effettuare il tirocinio), lo studente dovrà indicare il periodo nel quale vorrà effettuare l'attività ed attendere conferme da parte dell'azienda (prima) e dell'ufficio tirocini (poi).

L'**azienda**, tramite un proprio delegato, può richiedere il convenzionamento con l'Università: tale richiesta è quindi valutata da un impiegato dell'ufficio tirocini. In caso di esito positivo, l'azienda sarà in grado di caricare molteplici progetti formativi associati alle figure d'interesse per la stessa. **Tali progetti potranno essere scelti da più di uno studente** (si evince quindi un cambiamento nella prassi che prevede l'eliminazione dei progetti formativi ad-hoc in favore dell'introduzione di progetti formativi molto più simili a corsi aperti).

Un delegato aziendale prenderà in considerazione la domanda dello studente ed esaminandone il profilo ed il progetto formativo scelto, determinerà se questi possa risultare un innesto valido all'interno del contesto aziendale. Nel caso in cui la domanda venga accettata dall'azienda, questa verrà inoltrata all'ufficio tirocini per la validazione, passo che segnerà l'attivazione del tirocinio.

Sarà ovviamente un **impiegato dell'ufficio tirocini** a prendere in carico le domande di tirocinio: questi ne esaminerà il progetto formativo ed il profilo dello studente associato al fine di determinarne l'adeguatezza ed attivare il tirocinio.

Per ulteriori dettagli si consiglia di consultare la documentazione relativa alla prima versione del sistema relativamente alla fase di analisi e specifica dei requisiti, design, implementazione e testing reperibile al seguente link: <https://github.com/RCoccaro3/TirocinioSmart/tree/master/docs>.

4. Analisi delle modifiche proposte

Nella sua nuova versione, TirocinioSmart prevede una riorganizzazione della prassi che gli enti coinvolti dovranno seguire.

Punto cruciale nella manutenzione del sistema attuale, è rappresentata dall'aggiunta di un nuovo attore impersonificato dal Presidente del Consiglio Didattico che si inserirà nel sistema ed avrà il compito ultimo di validare una domanda di tirocinio giunta da uno studente.

Il sistema che si occupava precedentemente della gestione dei tirocini, infatti, prevedeva che lo studente consultasse l'elenco delle aziende convenzionate, effettuasse la domanda di tirocinio e che l'ufficio tirocini provvedesse alla raccolta della firma del Presidente del Consiglio Didattico.

Da qui è possibile intuire che il compito di validare una domanda di tirocinio non può essere lasciato al solo impiegato dell'ufficio tirocini bensì richiede anche la supervisione del Presidente del Consiglio Didattico. Pertanto, la seconda versione del sistema offre al Presidente del Consiglio Didattico la possibilità di validare o annullare definitivamente una domanda di tirocinio.

Ancora, la procedura per l'attivazione e lo svolgimento dei tirocini formativi prevede la compilazione di un Registro di Tirocinio durante l'attività presso la struttura ospitante. Dunque, saranno messe in campo un insieme di modifiche che consentirà, allo studente di poter redigere il proprio Registro di Tirocinio, inserendo volta per volta le attività svolte durante il tirocinio.

Il sistema, inoltre, consentirà all'azienda presso cui è svolto il tirocinio, la supervisione del registro elettronico garantendo la possibilità di esaminare le informazioni relative a ciascun'attività inserita e decidere se validare o annullare l'attività.

Allo stesso modo, il Presidente del Consiglio Didattico avrà l'opportunità di visionare il Registro elettronico di uno studente e validarlo; a questo punto, anche l'Ufficio Tirocini potrà accedere al sistema per esaminare il registro terminato e successivamente passare alla convalida dei cfu dal sistema Esse3 (che ricordiamo non comunica con il nostro sistema).

Infine, la piattaforma nella sua versione precedente, non prevede un sistema di notifiche che possa informare gli utenti interessati dell'avvenuto cambiamento di stato di un oggetto. A tal proposito, un'altra modifica riguarderà l'attivazione di un servizio di mailing invocato ogni qualvolta si verificherà un evento relativamente al quale un utente dovrà essere notificato: ad esempio, lo studente riceverà una mail nel momento in cui la sua domanda di tirocinio viene validata definitivamente dal Presidente del Consiglio Didattico.

4.1 Soluzione individuata

Prima di ogni cosa, il sistema dovrà conservare tutte le funzionalità già esistenti. Le modifiche apportate non avranno il compito di alterare l'attuale comportamento del sistema ma migliorare ed estendere le funzionalità già esistenti.

Priorità alta è associata alla Change Request relativa all'aggiunta del nuovo attore nel sistema in quanto garantire un'ulteriore validazione delle domande che giungono al Dipartimento di Informatica è contemplata nel regolamento come necessaria ai fini di una completa ed accurata attivazione del tirocinio stesso. In tal caso, l'insieme delle modifiche non implicheranno la realizzazione di un nuovo modulo poiché sarà necessario effettuare cambiamenti a classi e metodi già esistenti e aggiungerne dei nuovi non definiti in precedenza.

È dunque, per questa modifica proposta che riteniamo opportuno effettuare una buona impact analysis in modo da poter individuare package, classi e metodi coinvolti nell'implementazione della change request in questione.

Il sistema nasce come suddiviso in 3 layer; per ognuno di essi è stato realizzato un pacchetto, ciascuno dei quali contenente un discreto numero di sottosistemi, tranne che per il sottosistema di accesso ai dati persistenti. Questa divisione nasce dall'esigenza di dover riunire all'interno dello stesso pacchetto le classi che sono legate da una forte dipendenza da un punto di vista semantico.

Per tale ragione, per quanto riguarda la gestione del Registro di Tirocinio è prevista la realizzazione di un ulteriore sottosistema con il relativo pacchetto che possa raggruppare al suo interno le classi che lavorano e collaborano ai fini della concretizzazione del registro elettronico.

I sottosistemi esistenti non subiranno modifiche. Per tale ragione, quindi, la seconda change request non richiede ulteriore sforzo nella progettazione e realizzazione delle fasi che compongono il processo di Impact Analysis.

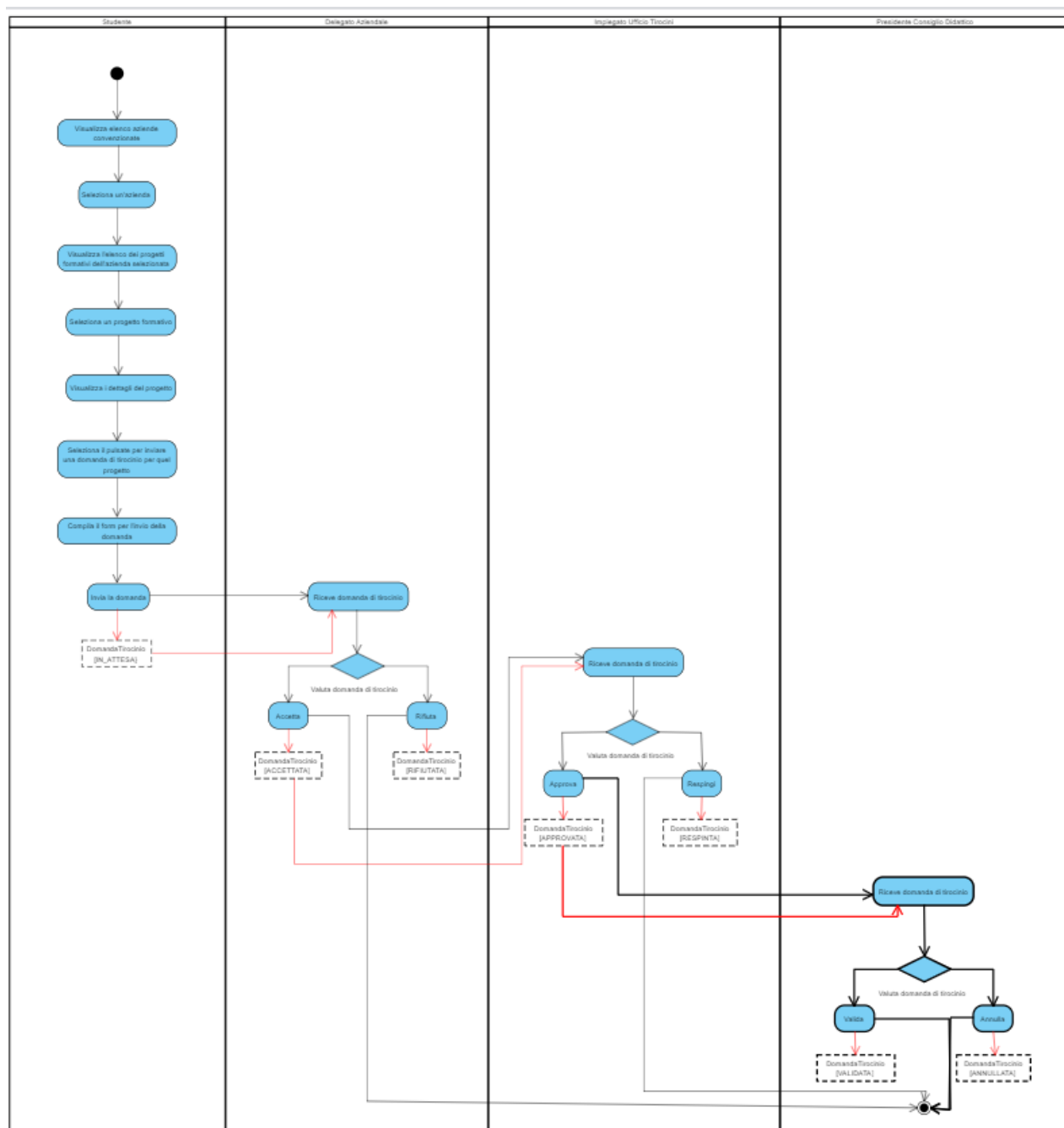
Conseguenza dell'implementazione della terza change request risulta essere la realizzazione di un quinto sottosistema che contenga al suo interno le classi che consentano di attivare un servizio di mailing. Sebbene la modifica richiesta provochi la definizione di ulteriore modulo, questo è il caso in cui i cambiamenti interessano anche altri moduli già esistenti; è pertanto necessario effettuare una buona analisi degli impatti per individuare i metodi che subiranno qualche leggera modifica per garantire la possibilità di inviare una mail.

4.2 Diagramma delle attività

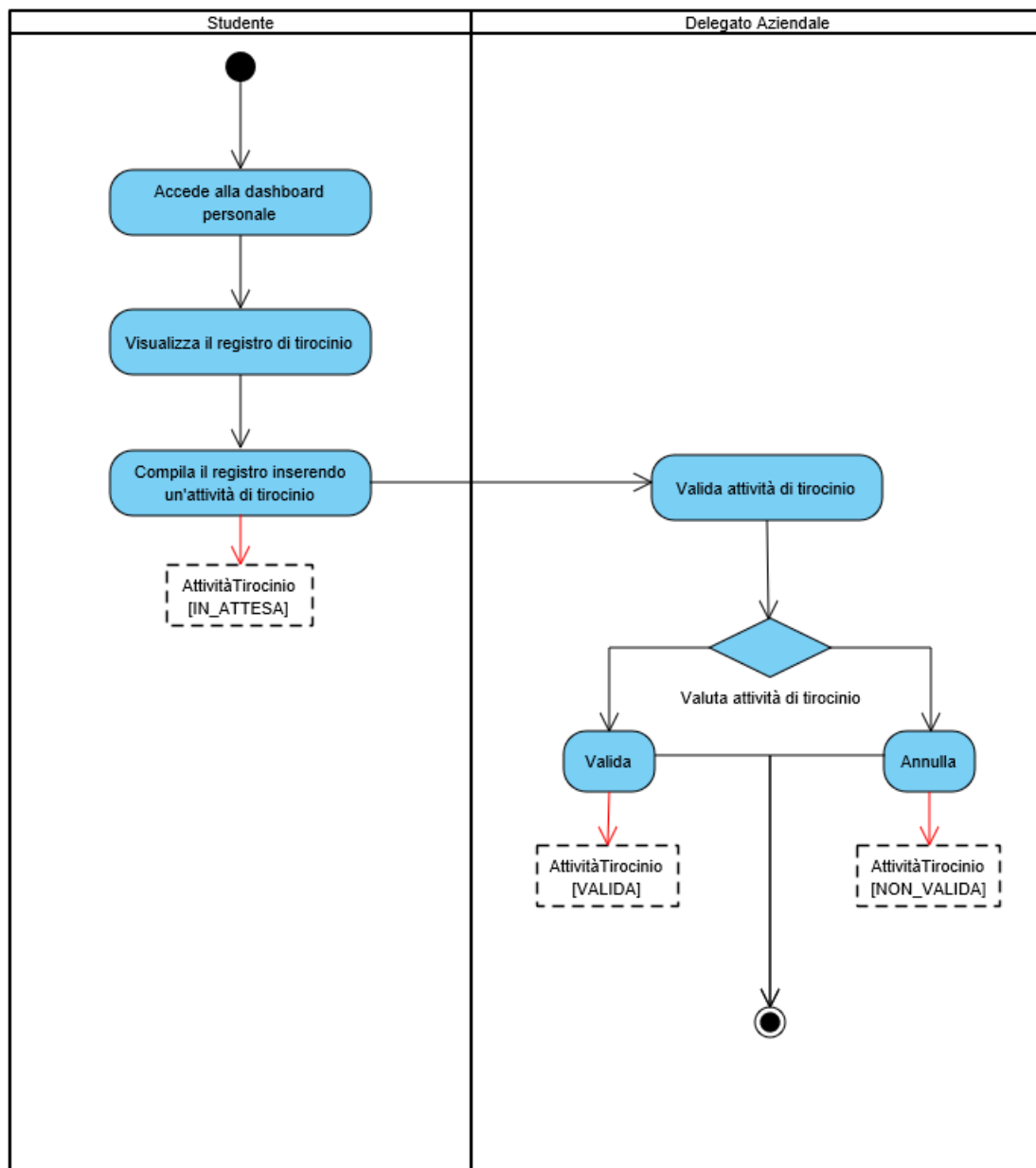
Al fine di poter cogliere al meglio gli effetti delle modifiche proposte al sistema per la gestione dei tirocini, di seguito vengono presentati gli activity diagram relativi alle nuove funzionalità introdotte.

In primis, vengono descritte le modifiche che subisce il processo di invio e convalida di una domanda di tirocinio: tali modifiche sono evidenziate all'interno del diagramma. Dopodiché, vengono presentate le modalità di gestione di un registro di tirocinio, evidenziando come avviene l'inserimento di una nuova attività e come, al termine del tirocinio, il registro venga chiusa e convalidato.

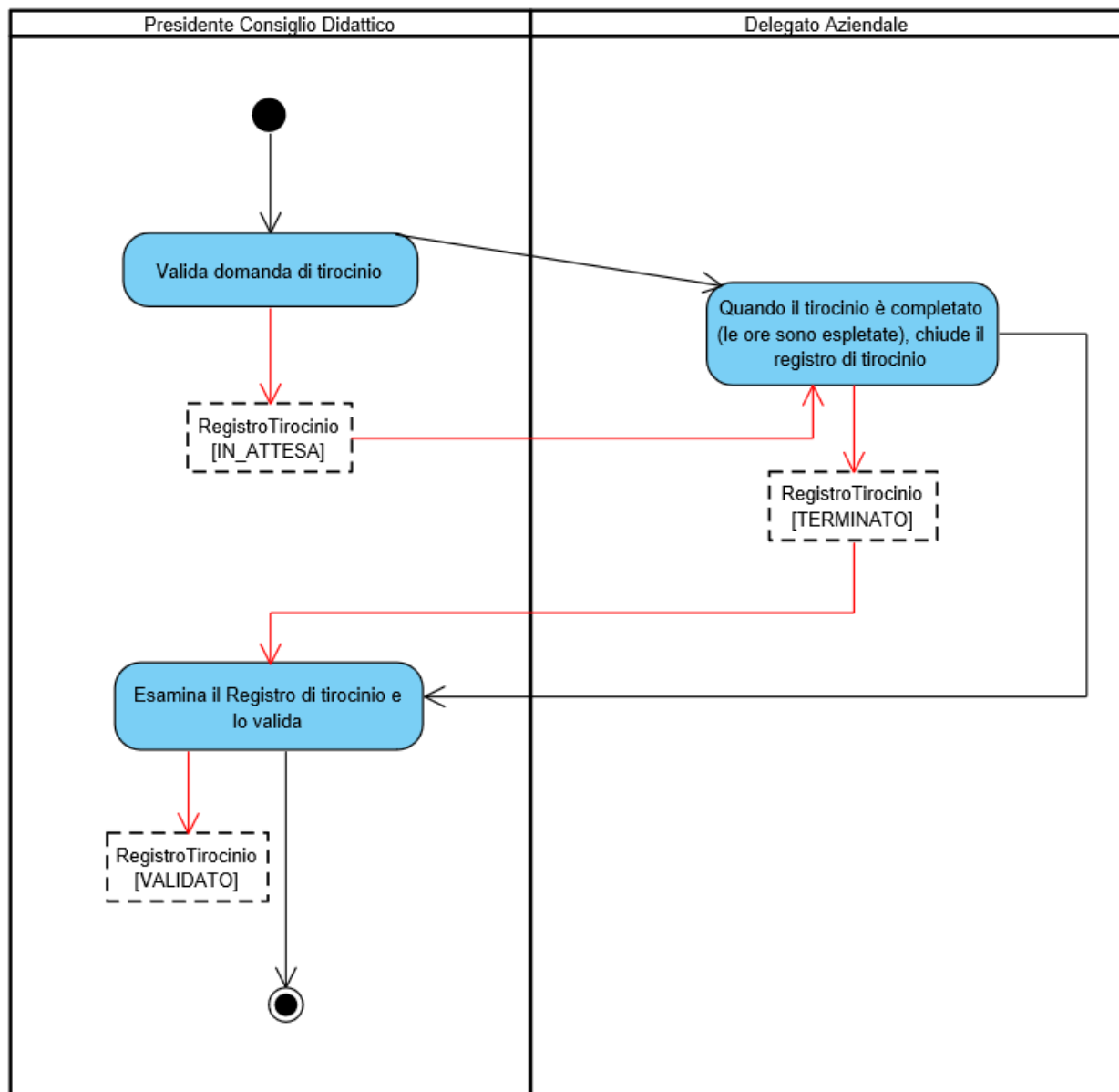
4.2.1 Invio di una domanda di tirocinio



4.2.2 Aggiunta nuova attività di tirocinio



4.2.3 Gestione e Convalida del Registro di tirocinio



4.3 Requisiti funzionali

Come definito per la prima versione del software, i requisiti funzionali sono stati organizzati in cinque categorie:

- requisiti per la gestione delle convenzioni;
- requisiti per la gestione dei progetti formativi;
- requisiti per la gestione delle domande di tirocinio;
- requisiti per la gestione dei tirocini;
- requisiti per la gestione dell'utenza.

L'implementazione della seconda change request comporta l'aggiunta di una nuova categoria di requisiti, ovvero, **“requisiti per la gestione dei registri di tirocinio”** di cui faranno parte tutti quei servizi che il sistema metterà a disposizione per garantire la compilazione e la convalida del registro di tirocinio.

Di seguito vengono elencati tutti requisiti del sistema, evidenziando, in grassetto, i nuovi requisiti che sono stati estrapolati dalle esigenze esposte nella descrizione delle change request.

4.3.1 Servizi per la gestione delle convenzioni

- RF_GC_1:RichiestaConvenzionamento
Il sistema dovrà fornire al delegato aziendale la possibilità di richiedere un convenzionamento
- RF_GC_2:ConsultazioneRichiesteConvenzionamento
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di consultare le richieste di convenzionamento in attesa
- RF_GC_3:ApprovazioneRichiestaConvenzionamento
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di approvare le richieste di convenzionamento
- RF_GC_4:RifiutoRichiestaConvenzionamento
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di rifiutare le richieste di convenzionamento fornendo una motivazione
- RF_GC_5:VisualizzazioneAziendeConvenzionate
Il sistema dovrà fornire all'utente la possibilità di consultare la lista delle aziende convenzionate
- RF_GC_6:VisualizzazioneAziendeSenzaBarriereConvenzionate
Il sistema dovrà fornire agli utenti la possibilità di consultare la lista delle aziende senza barriere architettoniche convenzionate

4.3.2 Requisiti per la gestione dei progetti formativi

- RF_GPF_1:VisualizzazioneProgettiFormativi
Il sistema dovrà fornire all'utente la possibilità di consultare la lista dei progetti formativi delle aziende convenzionate
- RF_GPF_2:VisualizzazioneDettaglioProgettoFormativo
Il sistema dovrà fornire all'utente la possibilità di consultare ogni singolo progetto formativo nel dettaglio
- RF_GPF_3:AggiuntaProgettoFormativo
Il sistema dovrà fornire al delegato aziendale la possibilità di aggiungere un nuovo progetto formativo a quelli dell'azienda
- RF_GPF_4:ArchiviazioneProgettoFormativo
Il sistema dovrà fornire al delegato aziendale la possibilità di archiviare un progetto formativo presente tra quelli dell'azienda

4.3.3 Requisiti per la gestione delle domande di tirocinio

- RF_GDT_1:InvioDomandaTirocinio
Il sistema dovrà offrire allo studente la possibilità di proporsi come tirocinante alle aziende, specificando il progetto formativo ed il periodo in cui condurre l'esperienza

- **RF_GDT_2:ConsultazioneStatoDomandeStudente**
Il sistema dovrà offrire allo studente la possibilità di consultare lo stato delle domande di tirocinio inviate
- **RF_GDT_3:ConsultazioneDomandeAzienda**
Il sistema dovrà offrire al delegato aziendale la possibilità di consultare le domande di tirocinio pervenute all'azienda che rappresenta
- **RF_GDT_4:AccettazioneDomandaTirocinio**
Il sistema dovrà offrire al delegato aziendale la possibilità di accettare le domande di tirocinio pervenute all'azienda che rappresenta
- **RF_GDT_5:RifiutoDomandaTirocinio**
Il sistema dovrà offrire al delegato aziendale la possibilità di rifiutare le domande di tirocinio pervenute all'azienda che rappresenta fornendo una motivazione
- **RF_GDT_6:ConsultazioneDomandeTirocinio**
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di visualizzare le domande di tirocinio accettate dalle aziende
- **RF_GDT_7:ApprovazioneDomandaTirocinio**
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di approvare le domande di tirocinio accettate dalle aziende
- **RF_GDT_8:RespintaDomandaTirocinio**
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di respingere le domande di tirocinio accettate dalle aziende fornendo una motivazione
- **RF_GDT_9:ConsultazioneDomandeInoltrateAzienda**
Il sistema dovrà fornire al delegato aziendale la possibilità di visualizzare l'elenco delle domande inoltrate all'ufficio tirocini con il relativo stato
- **RF_GDT_10:ConsultazioneDomandeRicevutePresidente**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di visualizzare l'elenco delle domande precedentemente approvate dall'Ufficio Tirocini
- **RF_GDT_11:ValidazioneDomandaTirocinio**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di validare le domande di tirocinio approvate dall'Impiegato dell'Ufficio Tirocini
- **RF_GDT_12:AnnullamentoDomandaTirocinio**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di annullare le domande di tirocinio approvate dall'Impiegato dell'Ufficio Tirocini
- **RF_GDT_13:ConsultazioneDomandeInoltrateUfficio**
Il sistema dovrà fornire all'Impiegato dell'Ufficio Tirocini la possibilità di visualizzare l'elenco delle domande inoltrate al Presidente del Consiglio Didattico con il relativo stato

4.3.4 Requisiti per la gestione dei tirocini

- **RF_GT_1:ConsultazioneTirociniStudente**
Il sistema dovrà fornire allo studente la possibilità di visualizzare i tirocini cui sta partecipando

- **RF_GT_2:ConsultazioneTirociniAzienda**
Il sistema dovrà fornire al delegato aziendale la possibilità di visualizzare l'elenco dei tirocini in corso legati alla sua azienda
- **RF_GT_3:ConsultazioneTirociniUfficio**
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di visualizzare l'elenco dei tirocini in corso di svolgimento da parte degli studenti iscritti alla piattaforma
- **RF_GT_4:ConsultazioneTirociniPresidente**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di visualizzare l'elenco dei tirocini in corso di svolgimento da parte degli studenti iscritti alla piattaforma

4.3.5 Servizi per la gestione dei registri di tirocinio

- **RF_GRT_1:ConsultazioneRegistroStudente**
Il sistema dovrà fornire allo studente la possibilità di visualizzare il proprio registro di tirocinio
- **RF_GRT_2:AggiuntaAttivitàTirocinio**
Il sistema dovrà fornire allo studente la possibilità di aggiungere una nuova attività al proprio registro di tirocinio
- **RF_GRT_3:ConsultazioneRegistroAzienda**
Il sistema dovrà fornire al Delegato Aziendale la possibilità di visualizzare il registro di tirocinio di uno studente/tirocinante e il relativo stato
- **RF_GRT_4:ValidazioneAttivitàTirocinio**
Il sistema dovrà fornire al Delegato Aziendale la possibilità di validare le attività di tirocinio inserite dallo studente nel proprio registro di tirocinio
- **RF_GRT_5:AnnullamentoAttivitàTirocinio**
Il sistema dovrà fornire al Delegato Aziendale la possibilità di annullare le attività di tirocinio inserite dallo studente nel proprio registro di tirocinio
- **RF_GRT_6:ChiusuraRegistroTirocinio**
Il sistema dovrà fornire al Delegato Aziendale la possibilità di chiudere il registro di tirocinio non appena il tirocinio risulta essere concluso
- **RF_GRT_7:ConsultazioneRegistroPresidente**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di visualizzare il registro di tirocinio di uno studente/tirocinante e il relativo stato
- **RF_GRT_8:ValidazioneRegistroTirocinio**
Il sistema dovrà fornire al Presidente del Consiglio Didattico la possibilità di validare il registro di tirocinio di uno studente/tirocinante
- **RF_GRT_9:ConsultazioneRegistroImpiegato**
Il sistema dovrà fornire all'Impiegato dell'Ufficio Tirocini la possibilità di visualizzare il registro di tirocinio di uno studente/tirocinante e il relativo stato

4.3.6 Requisiti per la gestione dell'utenza

- RF_GU_1:Autenticazione
Il sistema dovrà fornire all'ospite la possibilità di autenticarsi
- RF_GU_2:RegistrazioneAccountStudente
Il sistema dovrà fornire all'ospite la possibilità di richiedere la registrazione di un nuovo account studente
- RF_GU_3:ConsultazioneRichiesteIscrizione
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di visualizzare le richieste d'iscrizione alla piattaforma da parte degli studenti
- RF_GU_4:ApprovazioneRichiestaIscrizione
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di validare gli account degli utenti che hanno richiesto l'iscrizione alla piattaforma
- RF_GU_5:Logout
Il sistema dovrà fornire agli utenti registrati che si sono autenticati la possibilità di effettuare il logout
- RF_GU_6:RifiutoRichiestaIscrizione
Il sistema dovrà fornire all'impiegato dell'ufficio tirocini la possibilità di rifiutare le richieste d'iscrizione alla piattaforma provenienti dagli utenti

4.4 Modello di sistema

Per migliorare il livello di comprensione delle modifiche proposte al sistema attuale, di seguito sono proposti i principali use cases che descrivono le funzionalità che la nuova versione del software dovrà implementare.

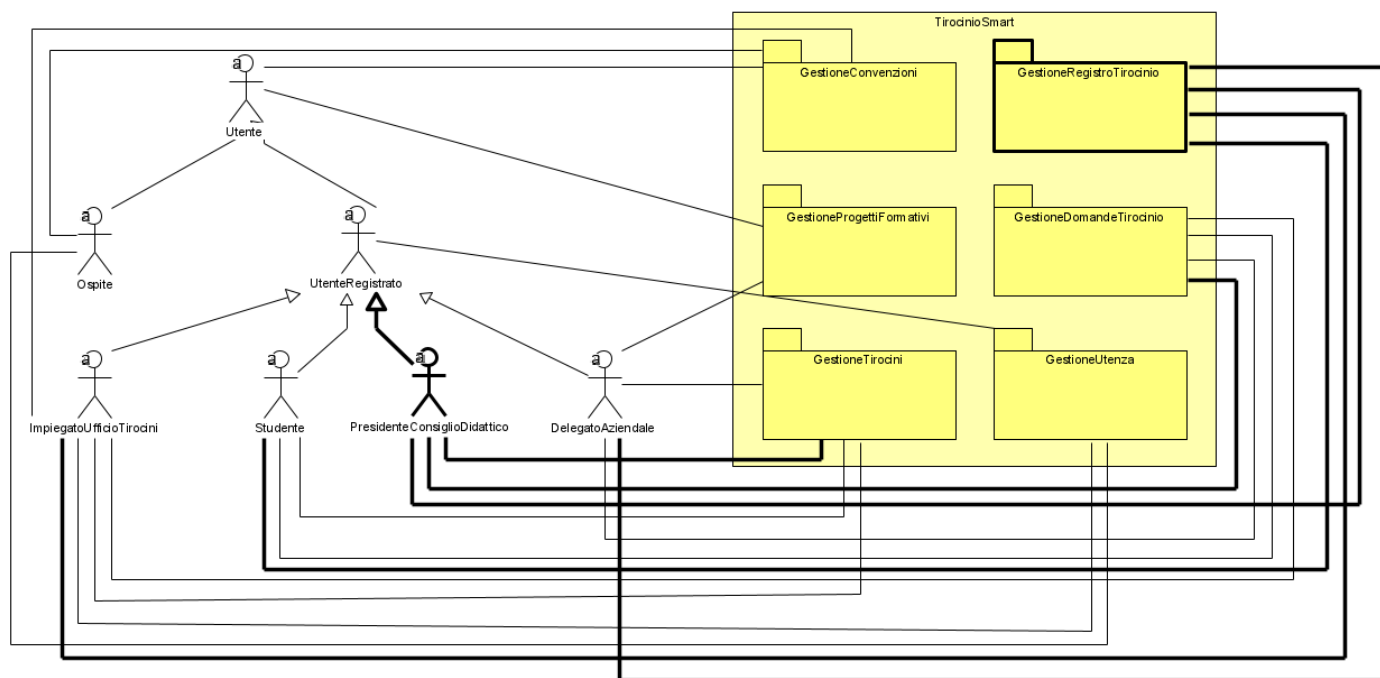
In generale, viene proposto lo use case diagram (aggiornato) che fornisce un'overview di tutto il sistema e chiarisce ulteriormente le estensioni proposte al sistema che risultano essere evidenziate.

Sottolineiamo, ancora una volta, che tutto ciò che risulta essere evidenziato all'interno degli use case diagram proposti di seguito fa parte della nuova versione del sistema e non del software già rilasciato.

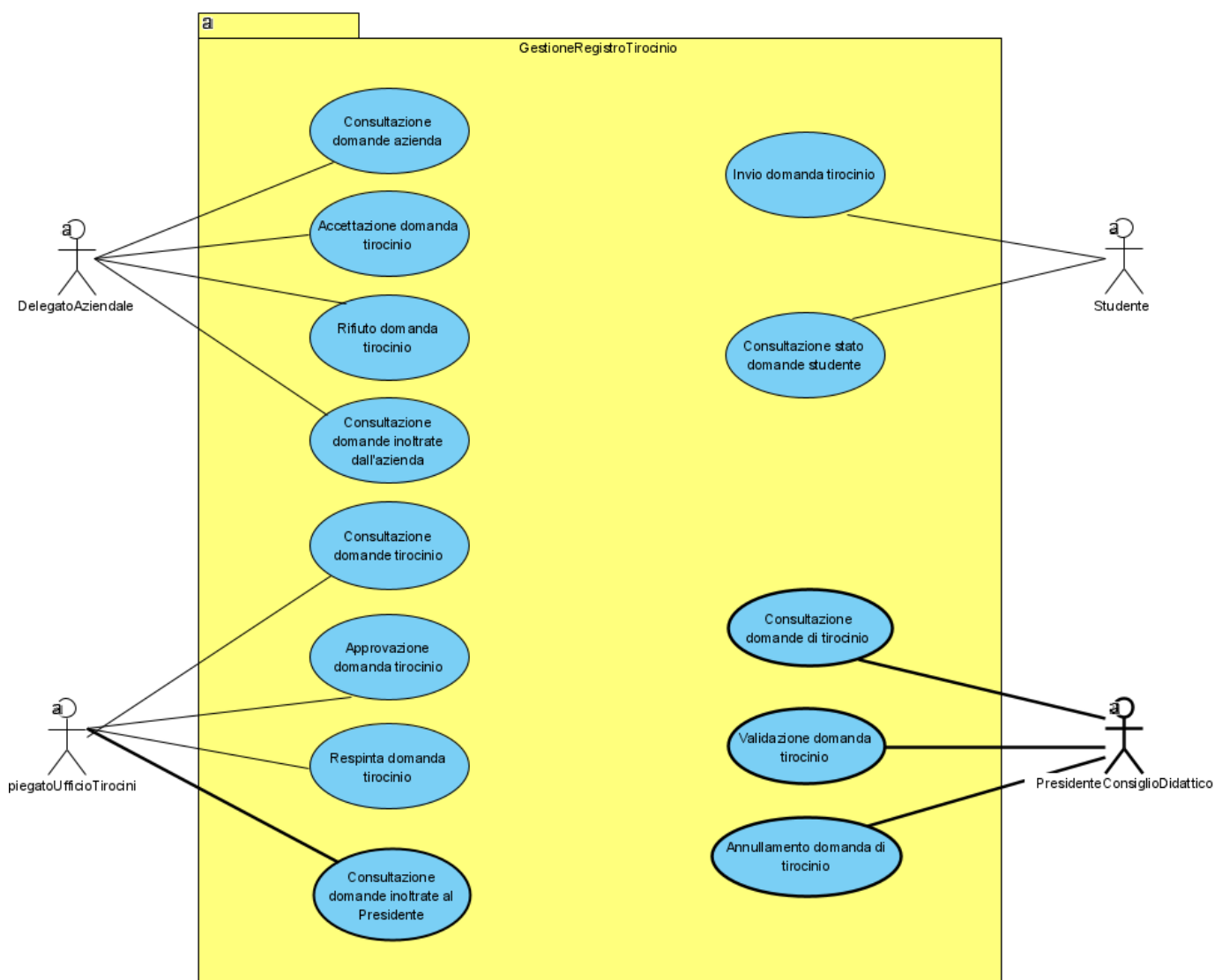
4.4.1 Use Case

Di seguito è mostrato lo use case diagram dell'intero sistema. Come possiamo notare tra utenti del sistema è inserito il Presidente del Consiglio Didattico che può operare effettuare operazioni sul Registro di Tirocinio, sui tirocini e sulle Domande di tirocinio.

Le modifiche apportate al sistema, inoltre, consentono sia al delegato che allo studente e all'impiegato la possibilità di usufruire delle nuove funzionalità per la gestione del registro di tirocinio elettronico.



4.4.1.1 UCD_GDT:Gestione domande tirocinio



Identificativo UC_GDT_10		ConsultazioneDomandeRicevutePresidente	Data	11/10/2020
			Versione	1.0
			Autore	RC
Descrizione		Lo use case definisce la funzionalità di visualizzazione della lista delle domande di tirocinio approvate dall’impiegato dell’ufficio tirocini e inoltrate al Presidente del Consiglio Didattico		
Attore Principale		Presidente Consiglio Didattico È interessato a visualizzare la lista delle domande di tirocinio approvate dall’ufficio tirocini		
Attori secondari		N/A		
Entry Condition		Il Presidente si è autenticato nel sistema ed è presentato il comando di visualizza- zione della lista delle domande di tirocinio		
Exit condition on success		Il sistema mostra la lista delle domande di tirocinio		
Exit condition on failure		Il sistema mostra un messaggio d’errore		
Rilevanza/User Priority		Elevata		
Frequenza stimata		40 usi/anno		
Extension point		N/A		
Generalization of		N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Presidente Consi- glio Didattico	Richiede la visualizzazione della lista delle domande di tirocinio approvate dall’ufficio tirocini tramite l’apposito comando		
2	Sistema	Mostra la lista delle domande di tirocinio		
Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista delle domande di tirocinio				
2.a1	Sistema	Mostra al Presidente del Consiglio Didattico un messaggio d’errore che lo informa del problema e lo invita a riprovare più tardi		

Identificativo UC_GDT_11	ValidazioneDomandaTirocinio	Data	11/08/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di validazione di una domanda di tirocinio approvata dall'ufficio tirocini		
Attore Principale	Presidente Consiglio Didattico È interessato a validare una domanda di tirocinio approvata dall'ufficio tirocini		
Attori secondari	N/A		
Entry Condition	Il Presidente si è autenticato nel sistema ed è visualizzata la lista delle domande di tirocinio approvate dall'ufficio tirocini		
Exit condition on success	Il sistema valida la domanda di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d'errore senza apportare modifiche alla domanda		

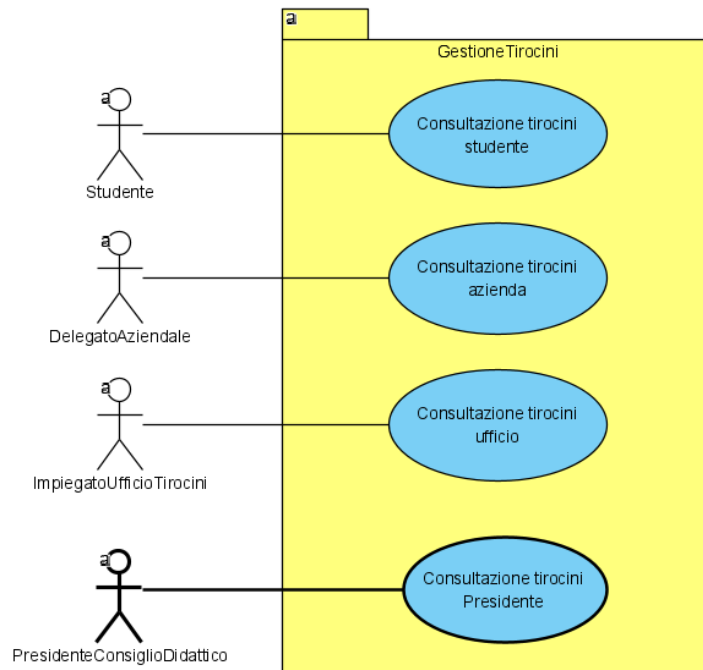
Rilevanza/User Priority		Elevata
Frequenza stimata		30 usi/anno
Extension point		N/A
Generalization of		N/A
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Presidente Consiglio Didattico	Richiede la validazione di una domanda di tirocinio tramite l'apposito comando
2	Sistema	Mostra un messaggio che richiede di confermare l'operazione
3	Presidente Consiglio Didattico	Conferma l'operazione
4	Sistema	Valida la domanda di tirocinio
5	Sistema	Conferma al Presidente del Consiglio Didattico che la procedura si è conclusa con successo
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio		
4.a1	Sistema	Mostra al presidente del consiglio didattico un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi

Identificativo UC_GDT_12		AnnullamentoDomandaTirocinio	Data	11/08/2020
			Versione	1.0
			Autore	RC
Descrizione		Lo use case definiscile la funzionalità di annullamento di una domanda di tirocinio approvata dall’ufficio tirocini		
Attore Principale		Presidente Consiglio Didattico È interessato a validare una domanda di tirocinio annullata dall’ufficio tirocini		
Attori secondari		N/A		
Entry Condition		Il Presidente si è autenticato nel sistema ed è visualizzata la lista delle domande di tirocinio approvate dall’ufficio tirocini		
Exit condition on success		Il sistema annulla la domanda di tirocinio		
Exit condition on failure		Il sistema mostra un messaggio d’errore senza apportare modifiche alla domanda		
Rilevanza/User Priority		Elevata		
Frequenza stimata		30 usi/anno		
Extension point		N/A		
Generalization of		N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Presidente Consiglio Didattico	Richiede l’annullamento di una domanda di tirocinio tramite l’apposito comando		
2	Sistema	Mostra un messaggio che richiede di confermare l’operazione		

3	Presidente Consiglio Didattico	Conferma l'operazione
4	Sistema	Annulla la domanda di tirocinio
5	Sistema	Conferma al Presidente del Consiglio Didattico che la procedura si è conclusa con successo
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio		
4.a1	Sistema	Mostra al presidente del consiglio didattico un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi

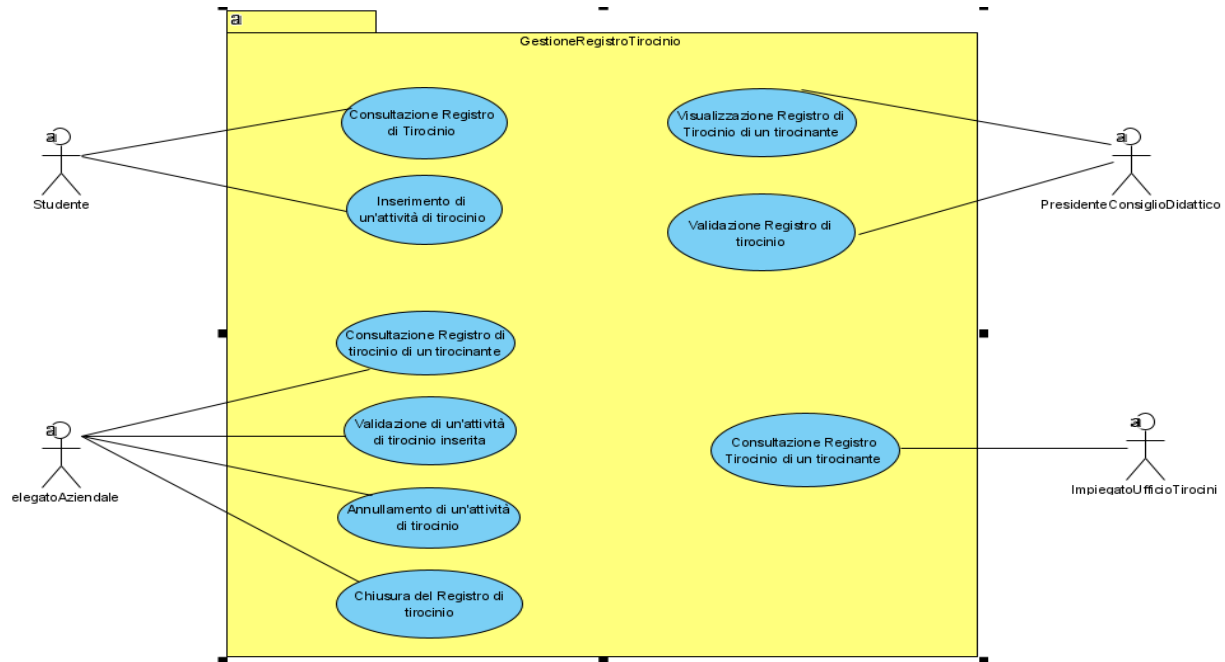
Identificativo UC_GDT_13		ConsultazioneDomandeInoltrateUfficio	Data	10/08/2020
			Versione	1.0
			Autore	RC
Descrizione		Lo use case definisce la funzionalità di visualizzazione delle domande di tirocinio già gestite dall’ufficio tirocini ed inoltrate al Presidente del Consiglio Didattico		
Attore Principale		Impiegato Ufficio Tirocini È interessato a visualizzare la lista delle domande di tirocinio approvate ed inoltrate al Presidente del Consiglio Didattico		
Attori secondari		N/A		
Entry Condition		L’Impiegato si è autenticato nel sistema ed è presentato il comando di visualizza- zione delle domande inoltrate al Presidente		
Exit condition on success		Il sistema mostra la lista delle domande inoltrate al Presidente del Consiglio Didat- tico		
Exit condition on failure		Il sistema mostra un messaggio d’errore		
Rilevanza/User Priority		Bassa		
Frequenza stimata		60 usi/anno		
Extension point		N/A		
Generalization of		N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Impiegato Ufficio Tirocini	Richiede la visualizzazione della lista delle domande di tirocinio attraverso l’apposito comando		
2	Sistema	Visualizza la lista delle domande di tirocinio inviate al Presidente		
Scenario/Flusso di eventi di ERRORE: il sistema non visualizza la lista delle domande di tirocinio				
2.e1	Sistema	Mostra all’impiegato un messaggio d’errore che segnala che non è stato possibile visualizzare la lista e lo invita a riprovare più tardi		

4.4.1.2 UCD_GT: Gestione tirocini



Identificativo UC_GT_4	ConsultazioneTirociniPresidente	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di visualizzazione dei tirocini in corso da parte del Presidente del Consiglio Didattico		
Attore Principale	Presidente del Consiglio Didattico È interessato a visualizzare la lista dei tirocini che sta svolgendo		
Attori secondari	N/A		
Entry Condition	Il Presidente del Consiglio Didattico si è autenticato nel sistema ed è mostrato il comando per la visualizzazione dei tirocini in corso		
Exit condition on success	Il sistema mostra al Presidente del Consiglio Didattico la lista dei tirocini in corso		
Exit condition on failure	Il sistema mostra un messaggio d'errore		
Rilevanza/User Priority	Media		
Frequenza stimata	60 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Presidente del Consiglio Didattico	Richiede la visualizzazione della lista dei tirocini in corso attraverso l'apposito comando	
2	Sistema	Visualizza la lista dei tirocini in corso	
Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista dei tirocini in corso			
2.a1	Sistema	Mostra al Presidente del Consiglio Didattico un messaggio d'errore che segnala che non è stato possibile visualizzare la lista e lo invita a riprovare più tardi	

4.4.1.2 UCD_GRT:Gestione registro tirocinio



Identificativo UC_GRT_1	ConsultazioneRegistroStudente	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di visualizzazione del registro di tirocinio da parte di uno studente		
Attore Principale	Studente È interessato a visualizzare la lista delle attività di tirocinio (registro di tirocinio) che sta svolgendo		
Attori secondari	N/A		
Entry Condition	Lo studente si è autenticato nel sistema ed è mostrato il comando per la visualizzazione del registro di tirocinio		
Exit condition on success	Il sistema mostra allo studente il proprio registro di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d'errore		
Rilevanza/User Priority	Media		
Frequenza stimata	60 usi/anno		
Extension point	N/A		
Generalization of	N/A		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Studente	Richiede la visualizzazione del registro di tirocinio attraverso l'apposito comando
2	Sistema	Visualizza la lista delle attività di tirocinio che compongono il registro di tirocinio

Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista dei tirocini in corso

2.a1	Sistema	Mostra allo studente un messaggio d'errore che segnala che non è stato possibile visualizzare il registro di tirocinio e lo invita a riprovare più tardi
------	---------	--

Identificativo UC_GRT_2	AggiuntaAttivitàTirocinio	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case descrive la funzionalità di aggiunta di un'attività di tirocinio al registro di tirocinio		
Attore Principale	Studente È interessato ad aggiungere una nuova attività al registro per il tirocinio in corso		
Attori secondari	N/A		
Entry Condition	Lo studente si è autenticato nel sistema ed è visualizzato il comando di aggiunta di un'attività di tirocinio		
Exit condition on success	Il sistema aggiunge l'attività di tirocinio alla lista delle attività relative a quello specifico registro di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d'errore senza apportare modifiche alla lista delle attività		
Rilevanza/User Priority	Elevata		
Frequenza stimata	20 usi/anno		
Extension point	N/A		
Generalization of	N/A		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Studente	Richiede di poter aggiungere un'attività di tirocinio tramite l'apposito comando
2	Sistema	Visualizza un modulo richiedente le seguenti informazioni (obbligatorie): <ul style="list-style-type: none"> • Nome dell'attività: stringa composta da 2 a 255 caratteri • Descrizione: stringa di almeno 2 caratteri • Numero ore dell'attività: intero compreso tra 1 e 12 • Orario di inizio dell'attività: <ul style="list-style-type: none"> ○ Ora di inizio: intero compreso tra 0 e 24 ○ Minuti di inizio: intero compreso tra 0 e 60 • Orario di fine: <ul style="list-style-type: none"> ○ Ora di fine: intero compreso tra 0 e 24 ○ Minuti di fine: intero compreso tra 0 e 60
3	Studente	Compila il modulo e lo invia
4	Sistema	Verifica che: <ul style="list-style-type: none"> • I campi rispettino i vincoli
5	Sistema	Salva i dati dell'attività inserita
6	Sistema	Conferma allo studente che l'inserimento dell'attività di tirocinio è avvenuto correttamente

Scenario/Flusso di eventi Alternativo: qualche campo obbligato non è stato riempito

5.a1	Sistema	Visualizza nuovamente il modulo contenente le informazioni inserite correttamente, mostrando però i messaggi di errore in corrispondenza dei campi vuoti
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad aggiungere un nuovo progetto formativo		
5.e1	Sistema	Mostra allo studente un messaggio d'errore che segnala che non è stato possibile salvare i dati e lo invita a riprovare più tardi

Identificativo UC_GRT_3	ConsultazioneRegistroAzienda	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di visualizzazione del registro di tirocinio da parte di un Delegato Aziendale		
Attore Principale	Delegato Aziendale È interessato a visualizzare la lista delle attività di tirocinio (registro di tirocinio) di uno studente/tirocinante		
Attori secondari	N/A		
Entry Condition	Il Delegato Aziendale si è autenticato nel sistema ed è mostrato il comando per la visualizzazione del registro di tirocinio		
Exit condition on success	Il sistema mostra al Delegato Aziendale il registro di tirocinio dello studente/tirocinante		
Exit condition on failure	Il sistema mostra un messaggio d'errore		
Rilevanza/User Priority	Media		
Frequenza stimata	60 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Delegato Aziendale	Richiede la visualizzazione del registro di tirocinio attraverso l'apposito comando	
2	Sistema	Visualizza la lista delle attività di tirocinio che compongono il registro di tirocinio	
Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista dei tirocini in corso			
2.a1	Sistema	Mostra al Delegato Aziendale un messaggio d'errore che segnala che non è stato possibile visualizzare il registro di tirocinio e lo invita a riprovare più tardi	

Identificativo UC_GRT_4	ValidazioneAttivitàTirocinio	Data	11/08/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di validazione di un'attività di tirocinio inserita da uno studente/tirocinante		
Attore Principale	Delegato Aziendale È interessato a validare un'attività di tirocinio inserita da uno studen-		

		te/tirocinante
Attori secondari		N/A
Entry Condition		Il Delegato Aziendale si è autenticato nel sistema ed è visualizzata la lista delle attività di tirocinio (registro di tirocinio) inserite dallo studente/tirocinante
Exit condition	on success	Il sistema valida l'attività di tirocinio
Exit condition	on failure	Il sistema mostra un messaggio d'errore senza apportare modifiche all'attività
Rilevanza/User Priority		Elevata
Frequenza stimata		30 usi/anno
Extension point		N/A
Generalization of		N/A
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Delegato Aziendale	Richiede la validazione di un'attività di tirocinio tramite l'apposito comando
2	Sistema	Mostra un messaggio che richiede di confermare l'operazione
3	Delegato Aziendale	Conferma l'operazione
4	Sistema	Valida l'attività di tirocinio
5	Sistema	Conferma al Delegato Aziendale che la procedura si è conclusa con successo
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio		
4.a1	Sistema	Mostra al delegato aziendale un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi

Identificativo UC_GRT_5	AnnullamentoAttivitàTirocinio	Data	11/08/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definiscile la funzionalità di annullamento di un’attività di tirocinio inserita da uno studente/tirocinante		
Attore Principale	Delegato Aziendale È interessato ad annullare un’attività di tirocinio inserita da uno studente/tirocinante		
Attori secondari	N/A		
Entry Condition	Il Delegato Aziendale si è autenticato nel sistema ed è visualizzata la lista delle attività di tirocinio (registro di tirocinio) inserite dallo studente/tirocinante		
Exit condition on success	Il sistema annulla l’attività di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d’errore senza apportare modifiche all’attività		
Rilevanza/User Priority	Elevata		
Frequenza stimata	30 usi/anno		

Extension point		N/A
Generalization of		N/A
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Delegato Aziendale	Richiede l'annullamento di un'attività di tirocinio tramite l'apposito comando
2	Sistema	Mostra un messaggio che richiede di confermare l'operazione
3	Delegato Aziendale	Conferma l'operazione
4	Sistema	Annulla l'attività di tirocinio
5	Sistema	Conferma al Delegato Aziendale che la procedura si è conclusa con successo
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio		
4.a1	Sistema	Mostra al delegato aziendale un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi

Identificativo UC_GRT_6	ChiusuraAttivitàTirocinio	Data	11/08/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definiscile la funzionalità di chiusura del registro di tirocinio redatto da uno studente/tirocinante		
Attore Principale	Delegato Aziendale È interessato a chiudere un registro di tirocinio redatto da uno studente/tirocinante		
Attori secondari	N/A		
Entry Condition	Il Delegato Aziendale si è autenticato nel sistema ed è visualizzato il registro di tirocinio (lista delle attività di tirocinio) redatto dallo studente/tirocinante		
Exit condition on success	Il sistema chiude il registro di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d’errore senza apportare modifiche al registro		
Rilevanza/User Priority	Elevata		
Frequenza stimata	30 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Delegato Aziendale	Richiede la chiusura del registro di tirocinio tramite l’apposito comando	
2	Sistema	Mostra un messaggio che richiede di confermare l’operazione	
3	Delegato Aziendale	Conferma l’operazione	
4	Sistema	Chiude il registro di tirocinio	
5	Sistema	Conferma al Delegato Aziendale che la procedura si è conclusa con successo	
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio			

4.a1	Sistema	Mostra al delegato aziendale un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi
------	---------	--

Identificativo UC_GRT_7	ConsultazioneRegistroPresidente	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di visualizzazione di un registro di tirocinio da parte del Presidente del Consiglio Didattico		
Attore Principale	Presidente del Consiglio Didattico È interessato a visualizzare la lista delle attività di tirocinio (registro di tirocinio) di uno studente/tirocinante		
Attori secondari	N/A		
Entry Condition	Il Presidente del Consiglio Didattico si è autenticato nel sistema ed è mostrato il comando per la visualizzazione del registro di tirocinio		
Exit condition on success	Il sistema mostra al Presidente del Consiglio Didattico il registro di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d'errore		
Rilevanza/User Priority	Media		
Frequenza stimata	60 usi/anno		
Extension point	N/A		
Generalization of	N/A		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Presidente del Consiglio Didattico	Richiede la visualizzazione del registro di tirocinio attraverso l'apposito comando	
2	Sistema	Visualizza la lista delle attività di tirocinio che compongono il registro di tirocinio	
Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista dei tirocini in corso			
2.a1	Sistema	Mostra al Presidente del Consiglio Didattico un messaggio d'errore che segnala che non è stato possibile visualizzare il registro di tirocinio e lo invita a riprovare più tardi	

Identificativo UC_GRT_8	ValidazioneRegistroTirocinio	Data	11/08/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di validazione di un registro di tirocinio redatto da uno studente/tirocinante		
Attore Principale	Presidente Consiglio Didattico È interessato a validare un registro di tirocinio redatto da uno studente/tirocinante		
Attori secondari	N/A		

Entry Condition	Il Presidente del Consiglio Didattico si è autenticato nel sistema ed è visualizzata la lista delle attività di tirocinio (registro di tirocinio) inserite dallo studente/tirocinante	
Exit condition on success	Il sistema valida il registro di tirocinio	
Exit condition on failure	Il sistema mostra un messaggio d'errore senza apportare modifiche al registro	
Rilevanza/User Priority	Elevata	
Frequenza stimata	30 usi/anno	
Extension point	N/A	
Generalization of	N/A	
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Presidente del Consiglio Didattico	Richiede la validazione di un registro di tirocinio tramite l'apposito comando
2	Sistema	Mostra un messaggio che richiede di confermare l'operazione
3	Presidente del Consiglio Didattico	Conferma l'operazione
4	Sistema	Valida il registro di tirocinio
5	Sistema	Conferma al Presidente del Consiglio Didattico che la procedura si è conclusa con successo
Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad approvare la domanda di tirocinio		
4.a1	Sistema	Mostra al Presidente del Consiglio Didattico un messaggio d'errore che lo informa del problema e lo invita a riprovare più tardi

Identificativo UC_GRT_9	ConsultazioneRegistroImpiegato	Data	11/10/2020
		Versione	1.0
		Autore	RC
Descrizione	Lo use case definisce la funzionalità di visualizzazione di un registro di tirocinio da parte di un Impiegato dell'Ufficio Tirocini		
Attore Principale	Impiegato Ufficio Tirocini È interessato a visualizzare la lista delle attività di tirocinio (registro di tirocinio) di uno studente/tirocinante		
Attori secondari	N/A		
Entry Condition	L' Impiegato dell'Ufficio Tirocini si è autenticato nel sistema ed è mostrato il comando per la visualizzazione del registro di tirocinio		
Exit condition on success	Il sistema mostra all' Impiegato dell'Ufficio Tirocini il registro di tirocinio		
Exit condition on failure	Il sistema mostra un messaggio d'errore		
Rilevanza/User Priority	Media		
Frequenza stimata	60 usi/anno		

Extension point		N/A
Generalization of		N/A
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Impiegato Ufficio Tirocini	Richiede la visualizzazione del registro di tirocinio attraverso l'apposito comando
2	Sistema	Visualizza la lista delle attività di tirocinio che compongono il registro di tirocinio
Scenario/Flusso di eventi di ERRORE: il sistema non riesce a caricare la lista dei tirocini in corso		
2.a1	Sistema	Mostra all' Impiegato dell'Ufficio Tirocini un messaggio d'errore che segnala che non è stato possibile visualizzare il registro di tirocinio e lo invita a riprovare più tardi

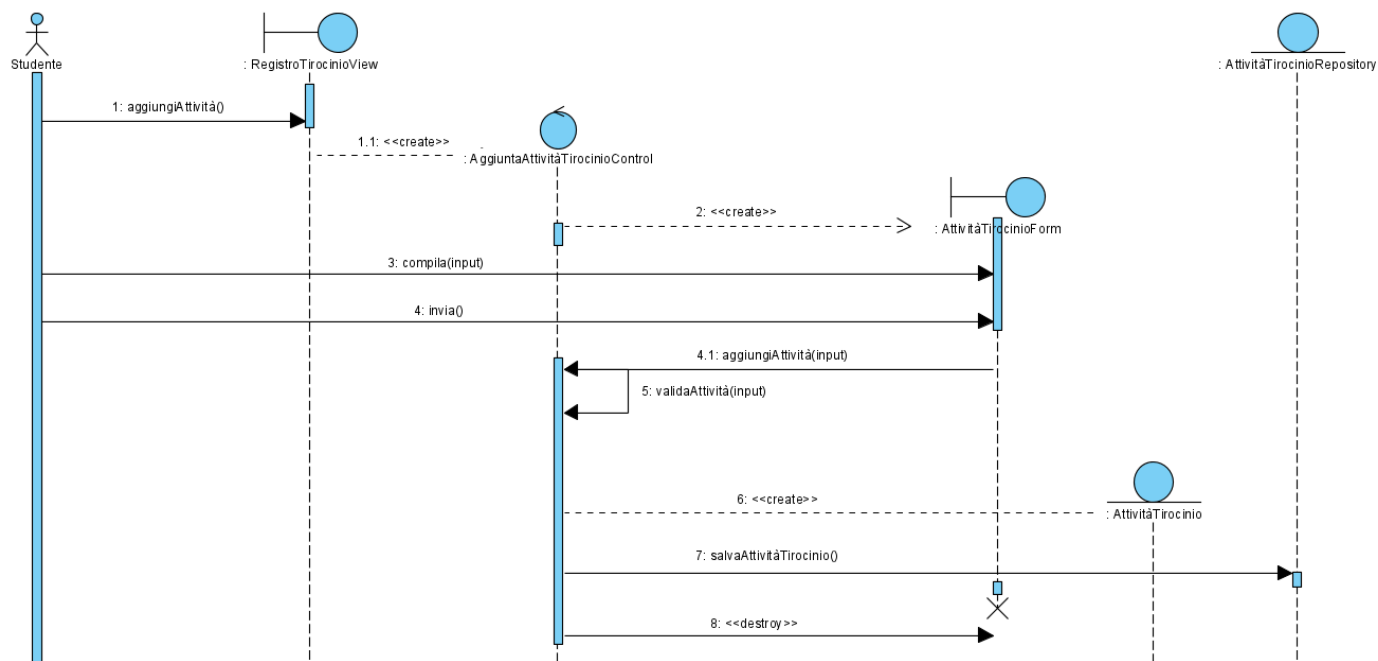
4.5 Modello dinamico

La descrizione del modello dinamico avviene tramite la rappresentazione delle interazioni tra l'utente ed il sistema (grazie all'utilizzo dei sequence diagrams) e la descrizione degli stati che caratterizzano il ciclo di vita degli oggetti (tramite l'utilizzo di state machine diagrams).

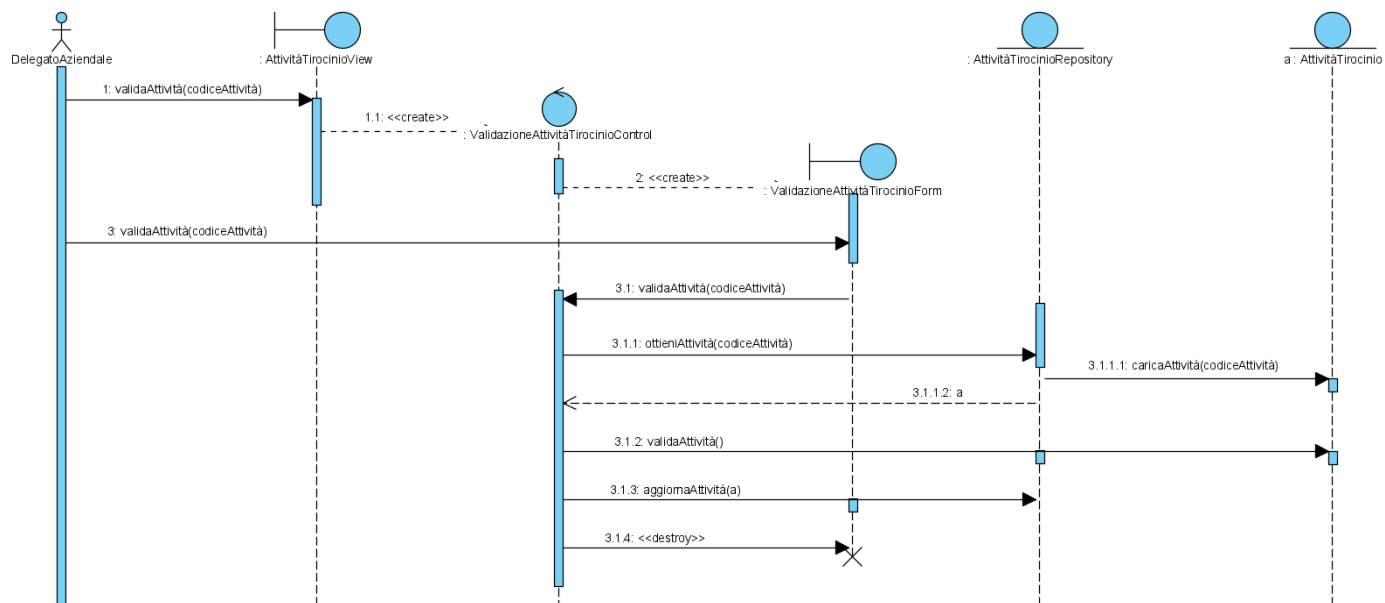
Sono omessi i diagrammi per i casi d'uso più banali e per gli oggetti i cui stati sono poco interessanti.

4.5.1 Sequence Diagram

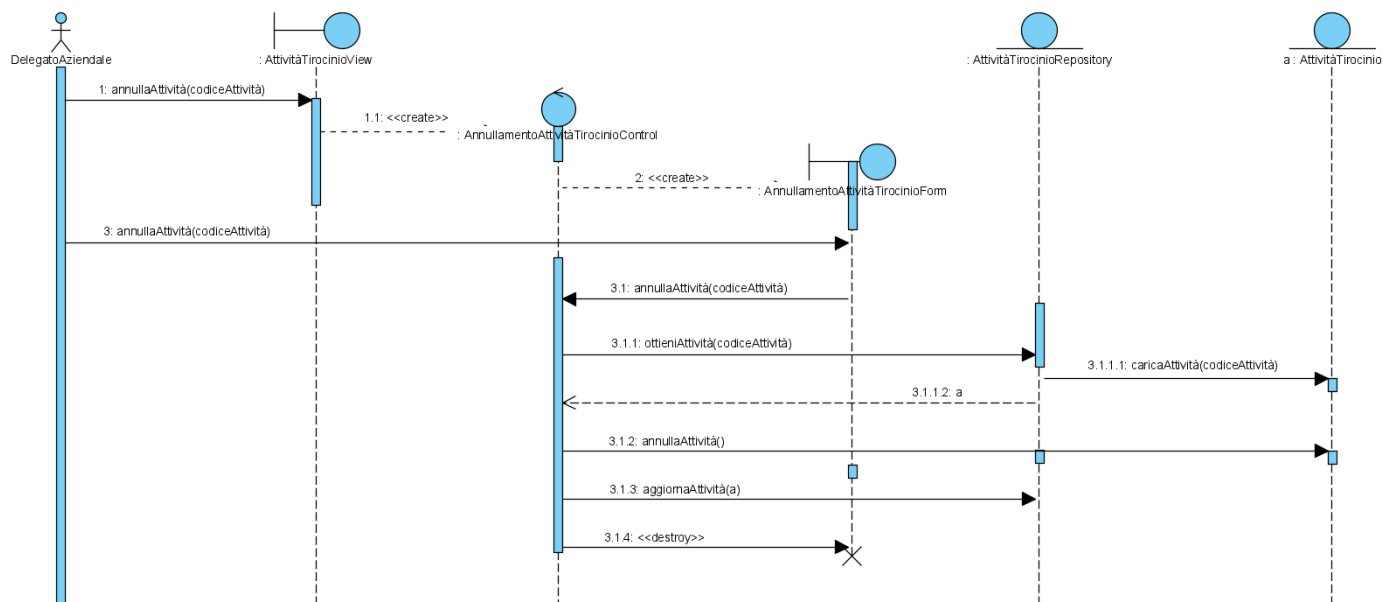
4.5.1.1 SD_GRT_2:AggiuntaAttivitàTirocinio



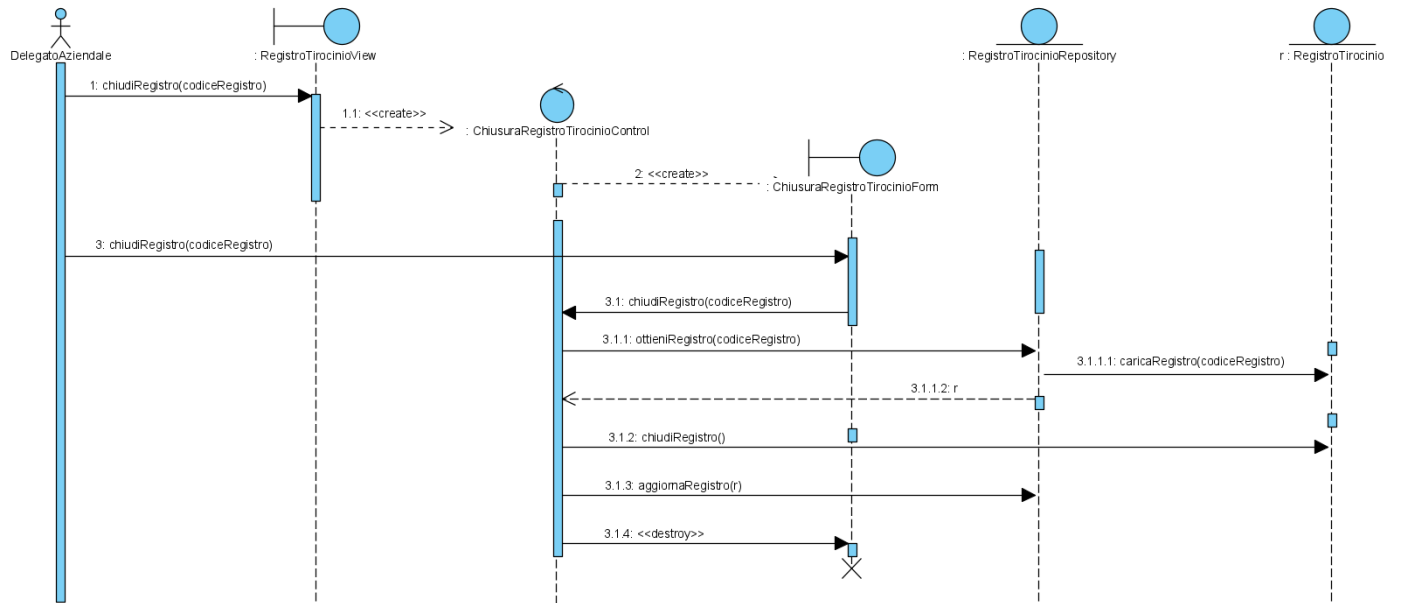
4.5.1.2 SD_GRT_4:ValidazioneAttivitàTirocinio



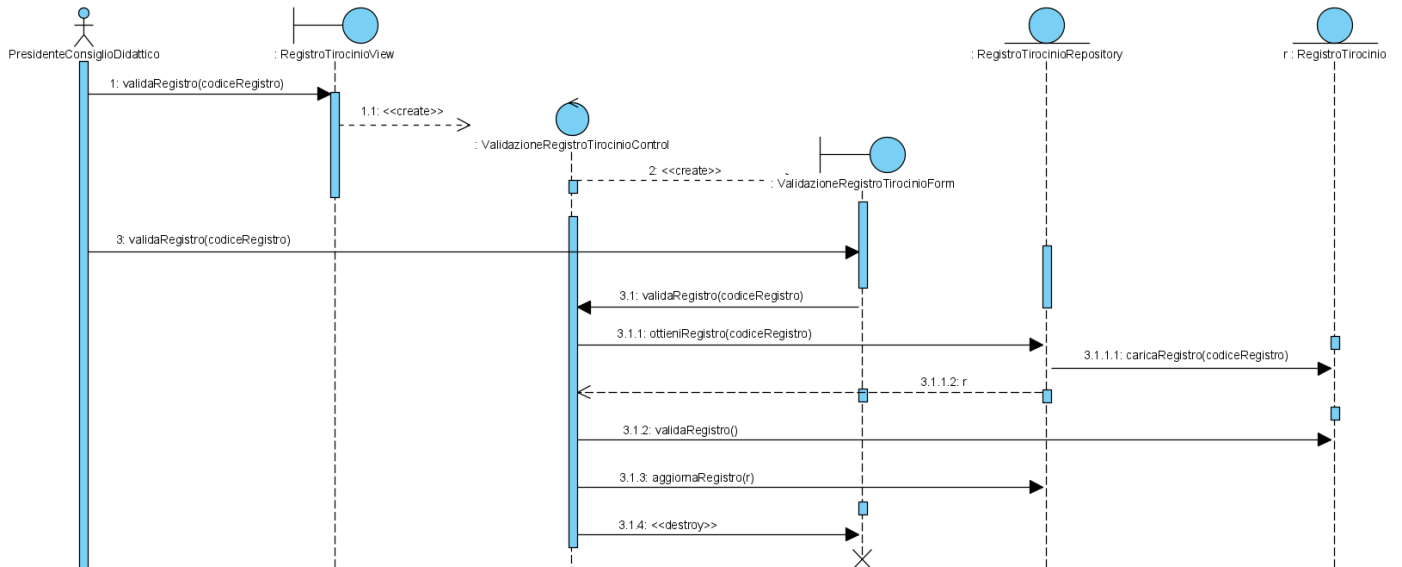
4.5.1.3 SD_GRT_5:AnnullamentoAttivitàTirocinio



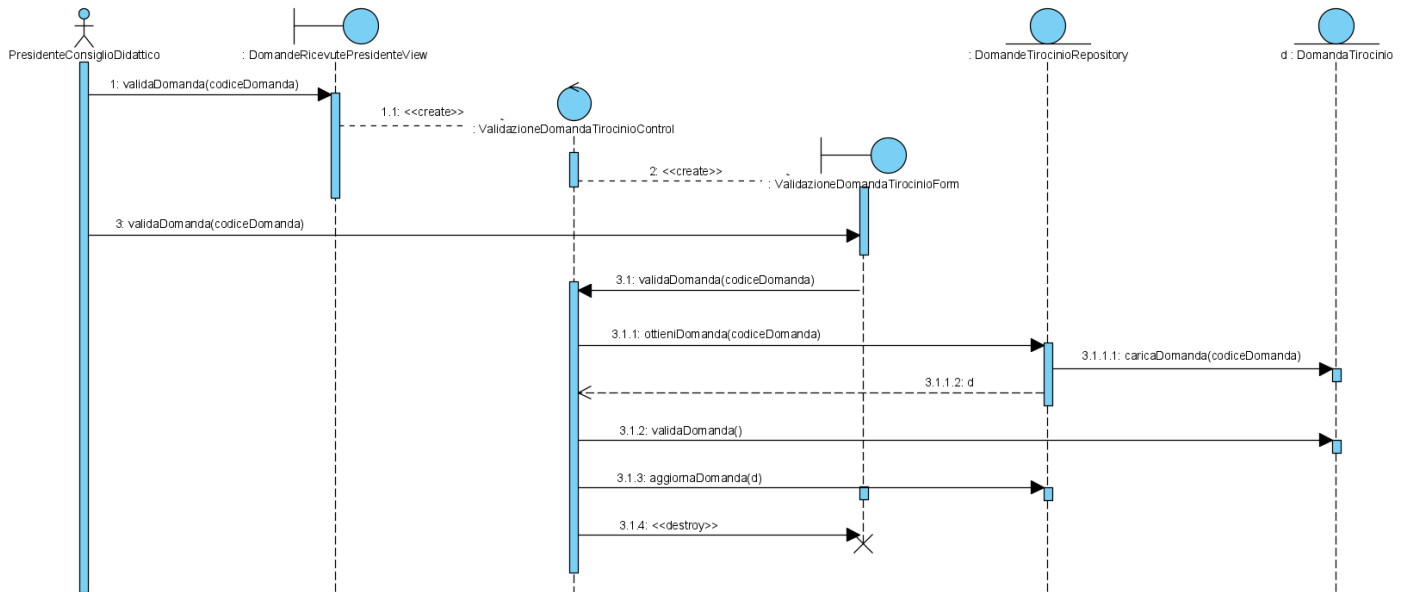
4.5.1.4 SD_GRT_6:ChiusuraRegistroTirocinio



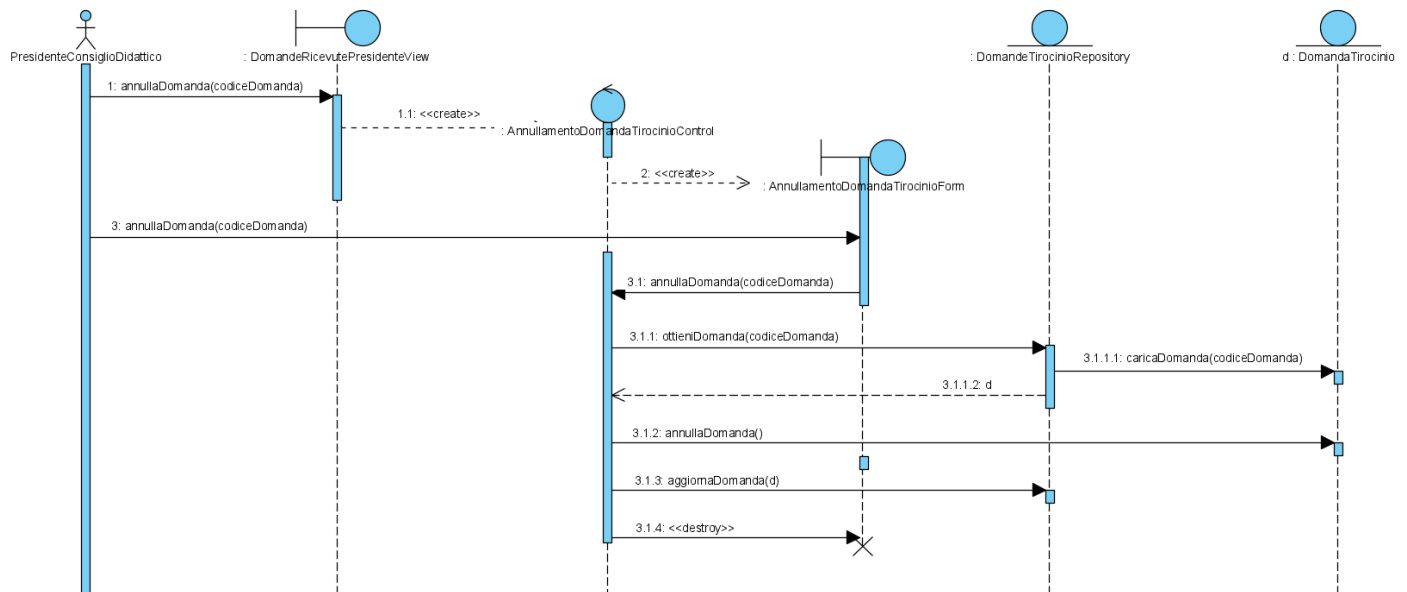
4.5.1.5 SD_GRT_8:ValidazioneRegistroTirocinio



4.5.1.6 SD_GDT_11:ValidazioneDomandaTirocinio



4.5.1.7 SD_GDT_12:AnnullamentoDomandaTirocinio

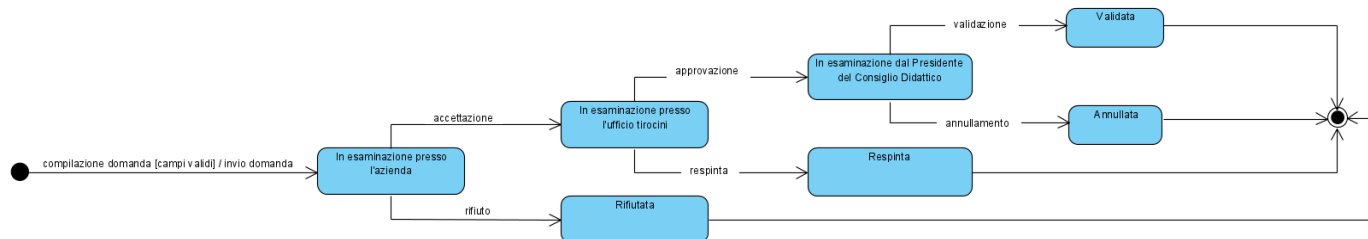


4.5.2 Statechart

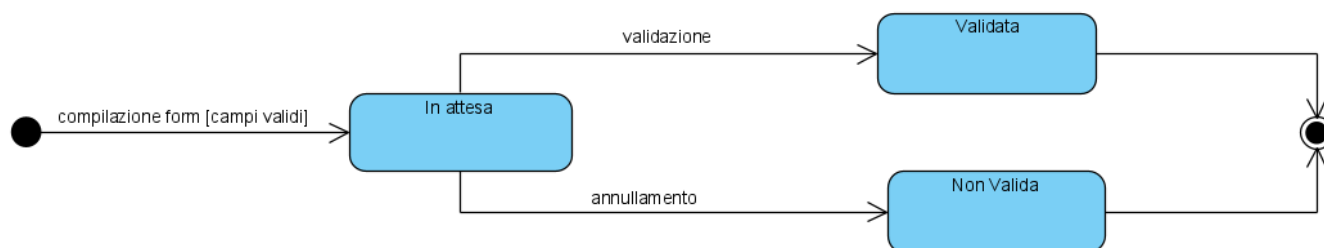
Nella sezione relativa agli statechart sono riportati i diagrammi degli stati relativi ai nuovi oggetti derivanti dall'implementazione delle change request e all'oggetto "Domanda di Tirocinio" che subisce modifiche consistenti nel suo processo di validazione.

Come già anticipato, l'implementazione del registro di tirocinio comporta l'impiego di due nuovi oggetti: *registro di tirocinio*, e *attività di tirocinio*. Specifichiamo che l'insieme delle attività di tirocinio forma il registro stesso

4.5.2.1 SCD_GDT:DomandaTirocinio



4.5.2.2 SCD_GRT:AttivitàTirocinio

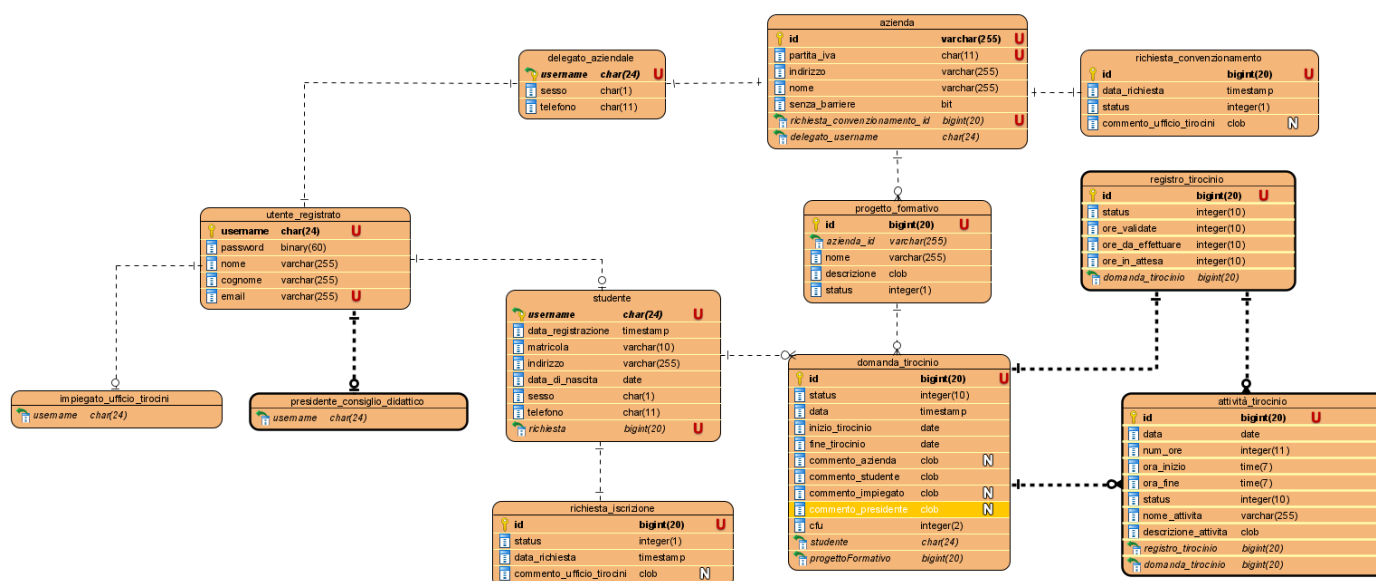


4.5.2.3 SCD_GRT:RegistroTirocinio



4.6 Gestione dei dati persistenti

Per la gestione dei dati persistenti, TirocinioSmart continua a fare affidamento su un database relazionale gestito tramite MySQL. Nel seguente schema sono evidenziate le tabelle e le relazioni che derivano dall'implementazione della seconda versione del software:



Come possiamo notare dallo schema ER, viene introdotta la tabella relativa alla figura del Presidente del Consiglio Didattico che eredita gli attributi dalla tabella “Utente Registrato”.

Inoltre, vengono definite le tabelle “registro_tirocinio” e “attività_tirocinio” utili ai fini delle operazioni relative alla concretizzazione del registro di tirocinio elettronico. In tal caso, possiamo notare che tra l’entità “Registro di Tirocinio” e “Attività di Tirocinio” esiste una relazione uno a molti, vale a dire che il registro di tirocinio si compone di un insieme di attività di tirocinio.

4.7 Decomposizione in sottosistemi

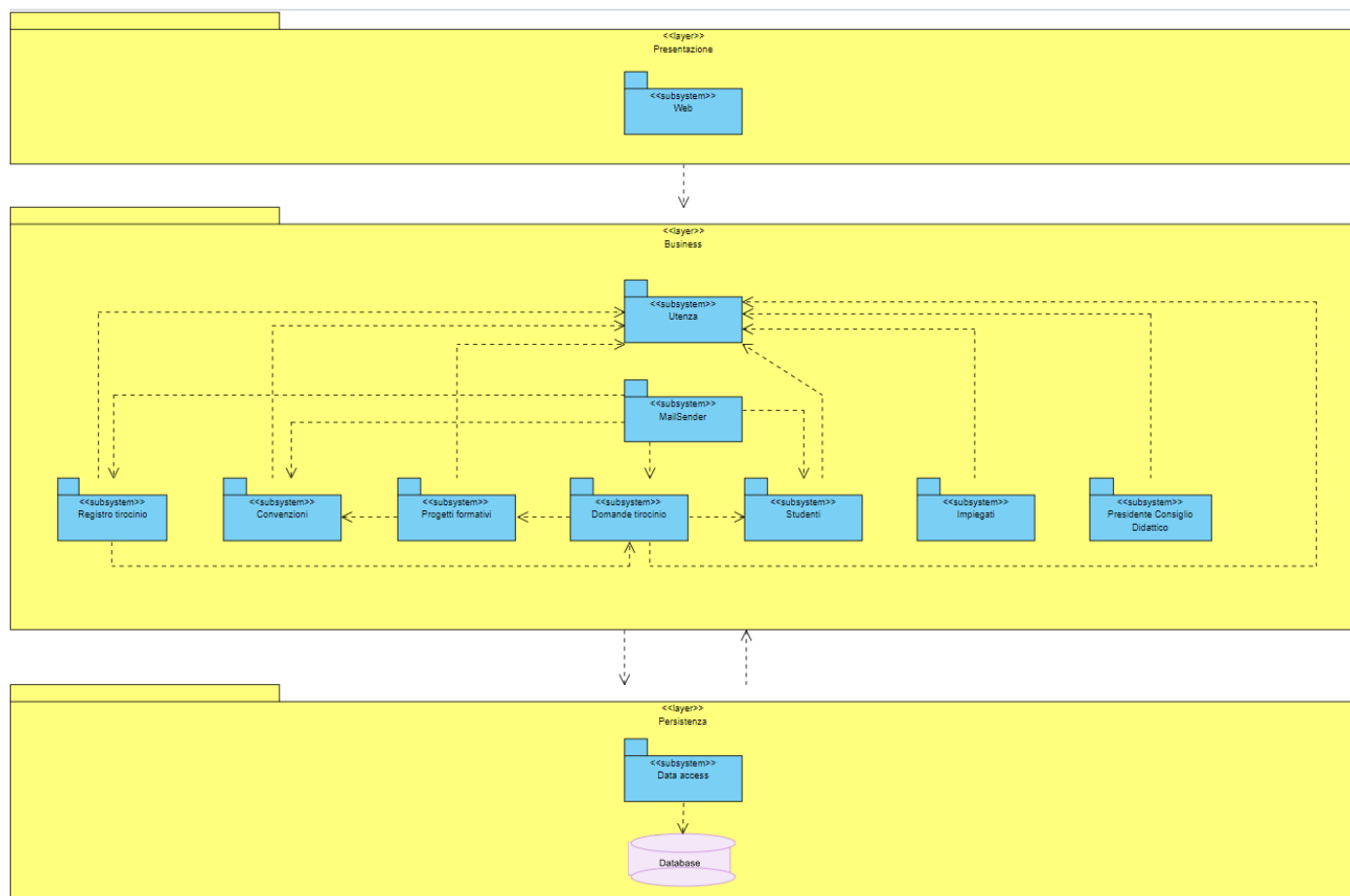
Il sistema risulta essere suddiviso in 3 livelli logici: presentazione, business e persistenza che si occupano rispettivamente di presentazione delle informazioni all’utente, definizione della logica applicativa e gestione dei dati persistenti.

Il livello di presentazione è composto da un solo sottosistema:

- **Web:** definisce l’interfaccia utente e si occupa del controllo del flusso di esecuzione.

Il livello di business, nella prima versione del sistema era composto da sei sottosistemi:

- **Utenza:** definisce l’utente generico del sistema ed offre tutti i servizi relativi all’autenticazione;



- **Convenzioni:** modella tutto ciò che riguarda il processo di convenzionamento e le aziende convenzionate in termini di entità e operazioni;
- **Studenti:** modella gli studenti ed il processo d’iscrizione alla piattaforma, definendo tutte le operazioni ad essi relative;

- **Impiegati:** modella gli impiegati dell'ufficio tirocini;
- **Progetti formativi:** modella i progetti formativi offerti dalle aziende e le relative operazioni;
- **Domande tirocinio:** modella le domande di tirocinio e le operazioni ad esse associate.

A questi sottosistemi, la nuova versione del software ne aggiunge altri tre:

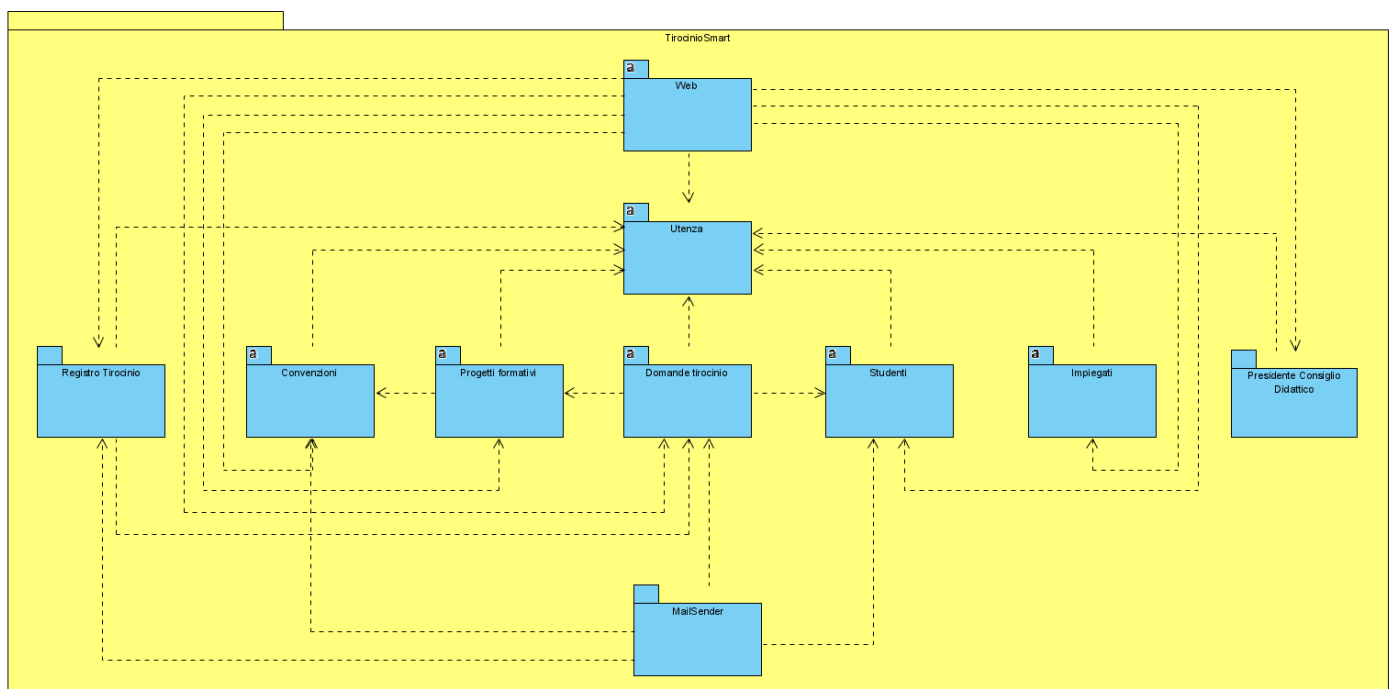
- **Registro tirocinio:** modella tutto ciò che concerne la gestione del registro di tirocinio di uno studente/tirocinante e le attività di tirocinio di cui esso si compone;
- **Presidente Consiglio Didattico:** modella le informazioni relative al Presidente del Consiglio Didattico;
- **MailSender:** definisce le operazioni che garantiscono l'esistenza del servizio di mailing.

Il livello di persistenza è invece composto dal sottosistema:

- **Data access:** si occupa del reperimento e del salvataggio delle informazioni manipolate dalla webapp dal/sul database sottostante.

Per ulteriori chiarimenti, si consiglia di consultare il documento di System Design relativo alla prima versione del software.

4.8 Divisione in pacchetti



La divisione del sistema in pacchetti condivide molti aspetti con la suddivisione in sottosistemi del paragrafo precedente. Infatti, come abbiamo già discusso, il sistema è suddiviso in tre layer ognuno dei quali contenente un certo numero di sottosistemi. La soluzione individuata è realizzare un pacchetto per ciascuno dei sottosistemi definiti tranne che per il sottosistema di accesso ai dati persistenti.

Ancora una volta, per ulteriori dettagli relativamente alle scelte prese nel suddividere il sistema in packages, è bene esaminare il documento di object design della prima versione del software.

4.9 Class Diagram

Questa sezione dovrebbe contenere il diagramma delle classi, ma date le dimensioni, l'immagine non sarebbe leggibile. Pertanto, una versione a dimensioni reali del diagramma in questione è disponibile in `docs/src/diagrammi/class/`.

5. Impact Analysis Process

La nuova versione del software si propone di apportare miglioramenti a quelle che sono le funzionalità già messe a disposizione dal sistema stesso; allo stesso tempo, le change request hanno come obiettivo anche quello di estendere il sistema, ed adeguarlo alle esigenze degli utenti, attraverso l'implementazione di nuovi requisiti funzionali (descritti in precedenza).

Dunque, l'analisi degli impatti si propone di rivelare le componenti, che nel nostro specifico caso sono rappresentate da classi Java, che risultano essere influenzate direttamente o indirettamente dalle modifiche proposte.

Il processo inizierà con l'individuazione dello **“Starting Impact Set”** (SIS). Dunque, in primo luogo, procediamo a identificare l'insieme iniziale di classi che potrebbero essere coinvolte direttamente nella modifica proposta. Questa fase consiste nell'analizzare il codice sorgente e le specifiche delle change request per poter selezionare le classi d'interesse della modifica.

Identificare il **“Candidate Impact Set”** (CIS) è il secondo step nel processo di impact analysis. In tal caso, incrementeremo il SIS con le componenti che molto probabilmente saranno modificate in seguito ai cambiamenti apportati agli elementi nel SIS.

Dopo aver individuato gli elementi che potenzialmente saranno coinvolti nelle modifiche al sistema, implementeremo la change request.

Successivamente, sarà possibile individuare l'**Actual Impact Set** (AIS) ovvero l'insieme delle componenti effettivamente modificate a seguito dell'implementazione della change request.

Individuare tali componenti ci permetterà di definire gli oggetti che non erano stati considerati come influenzati dalla modifica che però sono stati modificati durante la messa in opera della change request (**Discovered Impact Set**).

Potremo anche individuare gli elementi che faranno parte del **“False Positive Impact Set”** (FPIS) ovvero tutti gli elementi che erano stati identificati come influenzati dalla modifica ma che effettivamente non fanno parte dell'AIS.

Alla fine del processo di impact analysis saranno calcolate una serie di metriche per verificare la validità e l'efficienza del processo nelle fasi cruciali di selezione degli elementi probabilmente influenzati dalla modifica.

5.1 Individuazione dello Starting Impact Set (SIS)

In questa fase, procediamo ad identificare l'insieme iniziale delle classi che potrebbero essere coinvolte direttamente nella modifica proposta. Dunque, per definire lo Starting Impact Set, andiamo ad analizzare innanzitutto il testo delle due change request implicate nel processo di Impact Analysis ed il codice sorgente.

Nella tabella seguente vengono elencate le classi individuate come parte costitutiva dello Starting Impact Set. Per ciascuna delle componenti viene definito il grado con cui la modifica proposta impatta sulla componente stessa e una descrizione che si propone di illustrare il modo in cui la componente viene ristrutturata per accogliere l'implementazione della change request.

Il livello di impatto della modifica sulla componente è definito come:

- **High:** se la modifica comporta la completa riscrittura della classe;
- **Medium:** se la modifica comporta l'inserimento di nuovi metodi e/o la miglioira/ritocco di metodi già esistenti;
- **Low:** se la modifica comporta piccole correzioni ai metodi già esistenti.

Nei paragrafi successivi è definito lo Starting Impact Set per entrambe le change request che necessitano di analisi degli impatti. Ricordiamo che la change request relativa alla realizzazione del registro di tirocinio elettronico non necessita di impact analysis in quanto è definito un nuovo modulo per cui abbiamo a che fare con impatti davvero marginali sulle componenti già esistenti.

Precisiamo che le classi con il suffisso Service si occupano di realizzare la logica applicativa e le classi con il suffisso Controller si occupano di realizzare il controllo del flusso di esecuzione.

5.1.1 01-CR1:Aggiunta Attore

Componente	Impatto	Descrizione
DomandaTirocinio	Medium	La change request richiede che la Domanda di tirocinio sia ulteriormente validata. È necessario aggiungere due costanti intere che rappresentano i due nuovi stati: VALIDATA e ANNULLATA.
DomandeTirocinioService	Medium	La change request richiede l'aggiunta di funzionalità relativamente alla gestione delle Domande di Tirocinio. È pertanto necessario aggiungere nuovi metodi per garantire la validazione e l'annullamento delle domande di tirocinio; inoltre, è necessario modificare alcuni metodi già esistenti per consentire la visualizzazione delle domande di tirocinio inoltrate e/o ricevute e la visualizzazione dei tirocini in corso al nuovo attore del sistema.
UtenzaService	Medium	La change request prevede l'aggiunta di un nuovo attore al sistema al quale deve essere consentita la possibilità di autenticarsi nel sistema. È necessario quindi, modificare i metodi già esistenti che consentono l'autenticazione degli attori nel sistema.

5.1.2 03-CR3:Servizio di Mailing

Componente	Impatto	Descrizione
StudentiService	Medium	La change request richiede l'invio di una mail quando cambia lo stato di una Richiesta di iscrizione. È necessario modificare i metodi esistenti di invio, accettazione e rifiuto della richiesta chiamando il metodo che invoca il servizio di mailing.
DomandeTirocinioService	Medium	La change request richiede l'invio di una mail quando cambia lo stato di una Domanda di tirocinio. È necessario modificare i metodi esistenti di invio, accettazione, rifiuto, approvazione, respinta, validazione e annullamento della domanda chiamando il metodo che invoca il servizio di mailing.
ConvenzioniService	Medium	La change request richiede l'invio di una mail quando cambia lo stato di una Richiesta di Convenzionamento. È necessario modificare i metodi esistenti di invio, accettazione, rifiuto, approvazione e respinta della richiesta chiamando il metodo che invoca il servizio di mailing.

5.2 Individuazione del Candidate Impact Set (CIS)

Identificare il **“Candidate Impact Set”** (CIS) è il secondo step nel processo di impact analysis. In tal caso, ci occupiamo di incrementare lo Starting Impact Set con le componenti che molto probabilmente saranno modificate in seguito ai cambiamenti apportati agli elementi nel SIS.

Per fare ciò, è necessario analizzare il grafo delle dipendenze dal quale selezioniamo tutte le componenti che risultano essere dipendenti dalle classi già precedentemente inserite nell'impact set.

Il grafo delle dipendenze è stato generato automaticamente sulla prima versione del software attraverso l'utilizzo di un plug-in chiamato *Java Dependency Viewer*.

Successivamente, tra le classi estrapolate seguendo le dipendenze all'interno del grafo, si effettua una scelta per definire quelle che effettivamente è possibile siano impattate dalla modifica, al fine di ridurre l'insieme dei falsi positivi.

Di seguito, per ciascuna delle componenti individuate nello Starting Impact Set, è riportata una tabella che mostra tutte le classi che risultano dipendere da esse.

Classe	Dipendenze
DomandaTirocinio	DomandeTirocinioService, DomandaTirocinioRepository, ProgettoFormativo, Studente, DomandaTirocinioController
DomandeTirocinioService	DomandaTirocinioController
UtenzaService	DomandeTirocinioService, DomandaTirocinioController, UtenzaController, Au-

	tenticazioneInterceptor, ProgettoFormativoController, StudentiService, RichiestalscrizioneController, ProgettiFormativiService
StudentiService	RichiestalscrizioneController
ConvenzioniService	ConvenzioniController, ProgettiFormativiController

A questo punto, possiamo selezionare le classi Java con le quali ampliare lo Starting Impact Set e costruire il Candidate Impact Set.

Dunque, nei paragrafi seguenti è mostrato il Candidate Set per ognuna delle change request e l'insieme finale delle componenti che si presume siano impattate dalla modifica.

5.2.1 01-CR1: *Aggiunta Attore*

Precisiamo che le componenti evidenziate in grassetto risultano essere selezionate sulla base del grafo delle dipendenze.

Componente
DomandaTirocinio
DomandaTirocinioRepository
DomandeTirocinioService
DomandaTirocinioController
UtenzaService
UtenzaController
AutenticazioneHolder
AutenticazioneInterceptor

5.2.2 03-CR3: *Servizio di Mailing*

Precisiamo che le componenti evidenziate in grassetto risultano essere selezionate sulla base del grafo delle dipendenze.

Componente
StudentiService
RichiestalscrizioneController
ConvenzioniService
ConvenzioniController

DomandeTirocinioService
DomandaTirocinioController

5.2.3 Candidate Set finale

L'unione dei due Candidate Impact Set forma l'insieme finale delle componenti candidate ad essere modificati.

Componente
DomandaTirocinio
DomandaTirocinioRepository
DomandeTirocinioService
DomandaTirocinioController
UtenzaService
UtenzaController
AutenticazioneInterceptor
StudentiService
RichiestaIscrizioneController
ConvenzioniService
ConvenzioniController

Come possiamo notare, l'impact set è costituito da pochi elementi poiché il sistema è stato realizzato sin dall'inizio con l'obiettivo di ridurre al minimo le dipendenze tra le classi.

5.3 Definizione Actual Impact Set

In questa sezione approfondiamo i cambiamenti effettivi che si sono avuti durante l'implementazione delle modifiche e scopriamo le classi realmente impattate dalle migliorie.

Per ciascuna delle Change request vengono riportate le componenti impattate e una breve descrizione che illustra in che modo la componente stessa ha subito variazioni.

5.3.1 01-CR1:Aggiunta Attore

- **DomandaTirocinio:** La Domanda di tirocinio può avere due nuovi stati: è stato necessario aggiungere due costanti intere che rappresentano il caso in cui essa sia VALIDATA e il caso in cui essa sia ANNULLATA.


```

/**
 * Costante che rappresenta lo stato "validato" di una domanda di tirocinio.
 * Una domanda di tirocinio si trova in questo stato quando è stata gestita e validata
 * dal Presidente del Consiglio Didattico. <b>La domanda dev'essere stata precedentemente approvata</b>
 * per potervi assegnare questo stato.
 */
public static final int VALIDATA = 5;

/**
 * Costante che rappresenta lo stato "annullato" di una domanda di tirocinio.
 * Una domanda di tirocinio si trova in questo stato quando, sebbene approvata dall'Impiegato
 * dell'Ufficio Tirocini, la domanda è stata annullata dal Presidente del Consiglio Didattico.
 */
public static final int ANNULLATA = 6;

```

- **DomandaTirocinioService:** tale classe è modificata in quanto è stata necessaria l'aggiunta dei metodi per la validazione e l'annullamento di una domanda di tirocinio da parte del Presidente del Consiglio Didattico; inoltre è stato necessario modificare alcuni dei metodi esistenti per consentire la visualizzazione dei tirocini in corso e delle domande ricevute al Presidente del Consiglio Didattico e per consentire all'Impiegato la visualizzazione delle domande inoltrate.

```

/
@Transactional(rollbackFor = Exception.class)
public DomandaTirocinio validaDomandaTirocinio(long idDomanda)
    throws IdDomandaTirocinioNonValidoException,
           StatoDomandaNonIdoneoException,
           RichiestaNonAutorizzataException,
           DomandaTirocinioNonValidataException {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    // Solo un impiegato dell'ufficio tirocini può approvare una domanda di tirocinio
    if (!(utente instanceof PresidenteConsiglioDidattico)) {
        throw new RichiestaNonAutorizzataException();
    }

    // Controlla che la domanda esista
    DomandaTirocinio domanda = domandaRepository.findById(idDomanda);
    if (domanda == null) {
        throw new IdDomandaTirocinioNonValidoException();
    }

    if (domanda.getStatus() == DomandaTirocinio.APPROVATA) {
        domanda.setStatus(DomandaTirocinio.VALIDATA);

        //Una volta validata la domanda, associamo la domanda al registro
        //di tirocinio
        RegistroTirocinio registro = new RegistroTirocinio();
        registro.setDomandaTirocinio(domanda);
        domanda.setRegistroTirocinio(registro);
        registro = registroService.aggiungiRegistroTirocinio(registro);

        mailSender.sendEmail(domanda, domanda.getStudiante().getEmail());

        return domanda;
    } else {
        throw new StatoDomandaNonIdoneoException();
    }
}

```

```

@Transactional(rollbackFor = Exception.class)
public DomandaTirocinio annullaDomandaTirocinio(long idDomanda, String commento)
    throws IdDomandaTirocinioNonValidoException, StatoDomandaNonIdoneoException,
        CommentoDomandaTirocinioNonValidoException, RichiestaNonAutorizzataException {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    // Solo un impiegato dell'ufficio tirocini può respingere una domanda di tirocinio
    if (!(utente instanceof PresidenteConsiglioDidattico)) {
        throw new RichiestaNonAutorizzataException();
    }

    // Controlla che la domanda esista
    DomandaTirocinio domanda = domandaRepository.findById(idDomanda);
    if (domanda == null) {
        throw new IdDomandaTirocinioNonValidoException();
    }

    // Controlla che la richiesta non sia già stata gestita in precedenza ed impostare lo stato
    if (domanda.getStatus() == DomandaTirocinio.APPROVATA) {
        domanda.setStatus(DomandaTirocinio.ANNULLATA);
    } else {
        throw new StatoDomandaNonIdoneoException();
    }

    domanda.setCommentoPresidente(validaCommento(commento));
    mailSender.sendEmail(domanda, domanda.getStudente().getEmail());
    return domanda;
}

/
public List<DomandaTirocinio> elencaDomandeRicevute()
    throws RichiestaNonAutorizzataException {
    // Ottieni l'utente autenticato nel sistema
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    if (utente instanceof ImpiegatoUfficioTirocini) {
        // Un impiegato dell'ufficio tirocini può vedere solo le domande accettate
        return domandaRepository.findAllByStatus(DomandaTirocinio.ACCELTATA);
    } else if (utente instanceof DelegatoAziendale) {
        // Un delegato aziendale può vedere solo le domande in attesa
        DelegatoAziendale delegato = (DelegatoAziendale) utente;
        return domandaRepository.findAllByStatusAndProgettoFormativoAziendaId(
            DomandaTirocinio.IN_ATTESA,
            delegato.getAzienda().getId());
    } else if (utente instanceof Studente) {
        // Uno studente può vedere le domande di tirocinio in attesa
        Studente studente = (Studente) utente;
        return domandaRepository.findAllByStatusAndStudenteUsername(DomandaTirocinio.IN_ATTESA,
            studente.getUsername());
    } else if (utente instanceof PresidenteConsiglioDidattico) {
        return domandaRepository.findAllByStatus(DomandaTirocinio.APPROVATA);
    } else {
        throw new RichiestaNonAutorizzataException();
    }
}

```

```

public List<DomandaTirocinio> elencaTirociniInCorso(LocalDate giorno)
    throws RichiestaNonAutorizzataException {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    List<DomandaTirocinio> domandeApprovate;

    if (utente instanceof DelegatoAziendale) {
        DelegatoAziendale delegato = (DelegatoAziendale) utente;
        domandeApprovate = domandaRepository
            .findAllByStatusAndProgettoFormativoAziendaId(DomandaTirocinio.VALIDATA,
                delegato.getAzienda().getId());
    } else if (utente instanceof Studente) {
        Studente studente = (Studente) utente;
        domandeApprovate = domandaRepository
            .findAllByStatusAndStudenteUsername(
                DomandaTirocinio.VALIDATA,
                studente.getUsername());
    } else if (utente instanceof ImpiegatoUfficioTirocini) {
        domandeApprovate = domandaRepository.findAllByStatus(DomandaTirocinio.VALIDATA);
    } else if (utente instanceof PresidenteConsiglioDidattico) {
        domandeApprovate = domandaRepository.findAllByStatus(DomandaTirocinio.VALIDATA);
    } else {
        throw new RichiestaNonAutorizzataException();
    }

    List<DomandaTirocinio> tirociniInCorso = new ArrayList<DomandaTirocinio>();
    for (DomandaTirocinio domanda : domandeApprovate) {
        if ((domanda.getInizioTirocinio().isBefore(giorno)
            || domanda.getInizioTirocinio().isEqual(giorno))
            && domanda.getFineTirocinio().isAfter(giorno)) {
            tirociniInCorso.add(domanda);
        }
    }

    return tirociniInCorso;
}

else if (utente instanceof ImpiegatoUfficioTirocini) {
    List<DomandaTirocinio> domandeApprovate = domandaRepository
        .findAllByStatus(DomandaTirocinio.APPROVATA);

    List<DomandaTirocinio> domandeValidate = domandaRepository
        .findAllByStatus(DomandaTirocinio.VALIDATA);

    List<DomandaTirocinio> domandeAnnullate = domandaRepository
        .findAllByStatus(DomandaTirocinio.ANNULLATA);

    List<DomandaTirocinio> domandeInviate = new ArrayList<DomandaTirocinio>();
    domandeInviate.addAll(domandeValidate);
    domandeInviate.addAll(domandeApprovate);
    domandeInviate.addAll(domandeAnnullate);

    Collections.sort(domandeInviate);
}

```

**Nota: questa modifica è stata effettuata al metodo elencaDomandeInviate*

- **DomandaTirocinioController:** come specificato nel documento system design ed object design, il controller invoca i servizi del Service. Per tale ragione, è stato necessario implementare i due metodi per la validazione e l'annullamento di una domanda di tirocinio che invochino i metodi definiti all'interno della classe DomandaTirocinioService. Di conseguenza, è stato necessario, inoltre, effettuare alcune modifiche ai metodi di visualizzazione nel controller, corrispondenti ai metodi modificati all'interno della classe Service.

```

@RequestMapping(value = "/dashboard/domande/annulla", method = RequestMethod.POST)
public String annullaDomandaTirocinio(@RequestParam Long idDomanda,
                                      @RequestParam String commentoPresidente,
                                      RedirectAttributes redirectAttributes) {

    try {
        domandeService.annullaDomandaTirocinio(idDomanda, commentoPresidente);
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.domandaAnnullata");
    } catch (IdDomandaTirocinioNonValidoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.idNonValido");
    } catch (StatoDomandaNonIdoneoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.StatoDomandaNonIdoneo");
    } catch (CommentoDomandaTirocinioNonValidoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.commentoNonValido");
    } catch (RichiestaNonAutorizzataException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");
    } catch (Exception e) {
        logger.severe(e.getMessage());
        return "redirect:/errore";
    }

    return "redirect:/dashboard/domande/ricevute";
}

```

```

@RequestMapping(value = "/dashboard/domande/annulla", method = RequestMethod.POST)
public String annullaDomandaTirocinio(@RequestParam Long idDomanda,
                                      @RequestParam String commentoPresidente,
                                      RedirectAttributes redirectAttributes) {

    try {
        domandeService.annullaDomandaTirocinio(idDomanda, commentoPresidente);
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.domandaAnnullata");
    } catch (IdDomandaTirocinioNonValidoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.idNonValido");
    } catch (StatoDomandaNonIdoneoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.StatoDomandaNonIdoneo");
    } catch (CommentoDomandaTirocinioNonValidoException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.domandaTirocinio.commentoNonValido");
    } catch (RichiestaNonAutorizzataException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");
    } catch (Exception e) {
        logger.severe(e.getMessage());
        return "redirect:/errore";
    }

    return "redirect:/dashboard/domande/ricevute";
}

```

```

@RequestMapping(value = "/dashboard/tirocini", method = RequestMethod.GET)
public String visualizzaElencoTirociniInCorso(Model model,
                                             RedirectAttributes redirectAttributes) {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();
    List<DomandaTirocinio> tirociniInCorso;

    try {
        tirociniInCorso = domandeService.elencaTirociniInCorso();
        model.addAttribute("tirociniInCorso", tirociniInCorso);
    } catch (RichiestaNonAutorizzataException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");

        return "redirect:/";
    }

    if (utente instanceof ImpiegatoUfficioTirocini) {
        return "tirocini-in-corso-ufficio";
    } else if (utente instanceof DelegatoAziendale) {
        return "tirocini-in-corso-azienda";
    } else if (utente instanceof Studente) {
        return "tirocini-in-corso-studente";
    } else if (utente instanceof PresidenteConsiglioDidattico) {
        return "tirocini-in-corso-presidente";
    } else {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");

        return "redirect:/";
    }
}

```

```

/
@RequestMapping(value = "/dashboard/domande/inviare", method = RequestMethod.GET)
public String visualizzaElencoDomandeInviare(Model model, RedirectAttributes redirectAttributes) {
    try {
        List<DomandaTirocinio> domandeInviare = domandeService.elencaDomandeInviare();
        model.addAttribute("elencoDomandeInviare", domandeInviare);
    } catch (RichiestaNonAutorizzataException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");

        return "redirect:/";
    }

    if (utenzaService.getUtenteAutenticato() instanceof DelegatoAziendale) {
        return "domande-inviare-azienda";
    } else if (utenzaService.getUtenteAutenticato() instanceof Studente) {
        return "domande-inviare-studente";
    } else if (utenzaService.getUtenteAutenticato() instanceof ImpiegatoUfficioTirocini) {
        return "domande-inviare-ufficio";
    } else {
        redirectAttributes.addFlashAttribute("testoNotifica",
                                           "toast.autorizzazioni.richiestaNonAutorizzata");

        return "redirect:/";
    }
}

```

```

@RequestMapping(value = "/dashboard/domande/ricevute", method = RequestMethod.GET)
public String visualizzaElencoDomande(RedirectAttributes redirectAttributes, Model model) {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    try {
        List<DomandaTirocinio> domande = domandeService.elencaDomandeRicevute();
        model.addAttribute("elencoDomandeTirocinio", domande);
    } catch (RichiestaNonAutorizzataException e) {
        redirectAttributes.addFlashAttribute("testoNotifica",
            "toast.autorizzazioni.richiestaNonAutorizzata");

        return "redirect:/";
    }

    if (utente instanceof ImpiegatoUfficioTirocini) {
        return "domande-ricevute-ufficio";
    } else if (utente instanceof DelegatoAziendale) {
        return "domande-ricevute-azienda";
    } else if (utente instanceof PresidenteConsiglioDidattico) {
        return "domande-ricevute-presidente";
    }

    return "redirect:/errore";
}

```

- **UtenzaService:** tale classe espone i servizi necessari all'autenticazione degli utenti nel sistema. Siccome al nuovo utente è fornita la possibilità di loggarsi nella piattaforma, è stato necessario effettuare alcune modifiche ai metodi già esistenti quali:

```

public void login(String username, String password)
    throws CredenzialiNonValideException, RichiestaConvenzionamentoInAttesaException,
        RichiestaConvenzionamentoRifiutataException, RichiestaIscrizioneInAttesaException,
        RichiestaIscrizioneRifiutataException {
    UtenteRegistrato utente;

    utente = impiegatoRepository.findByUsernameAndPassword(username, password);
    if (utente != null) {
        AutenticazioneHolder.setUtente(username);
        return;
    }

    utente = presidenteRepository.findByUsernameAndPassword(username, password);
    if (utente != null) {
        AutenticazioneHolder.setUtente(username);
        return;
    }
}

```



```

public UtenteRegistrato getUtenteAutenticato() {
    // Ottieni l'username dell'utente autenticato e restituisci null se non è pres
    // in sessione
    String username = (AutenticazioneHolder.getUtente());
    if (username == null) {
        return null;
    }

    UtenteRegistrato utente;

    // Controlla se l'username è associato ad un impiegato dell'ufficio tirocini
    utente = impiegatoRepository.findByUsername(username);
    if (utente != null) {
        return utente;
    }

    // Controlla se l'username è associato ad un delegato aziendale
    utente = delegatoRepository.findByUsername(username);
    if (utente != null) {
        return utente;
    }

    // Controlla se l'username è associato ad uno studente
    utente = studenteRepository.findByUsername(username);
    if (utente != null) {
        return utente;
    }

    // Controlla se l'username è associato al Presidente
    utente = presidenteRepository.findByUsername(username);
    if (utente != null) {
        return utente;
    }

    return null;
}

```

- **UtenzaController:** modificato il Service, è necessario modificare il controller andando ad aggiungere qualche riga di codice al metodo che consente di visualizzare la propria dashboard dopo aver effettuato il login

```

@RequestMapping(value = "/dashboard", method = RequestMethod.GET)
public String mostraDashboard(RedirectAttributes redirectAttributes) {
    UtenteRegistrato utente = utenzaService.getUtenteAutenticato();

    if (utente == null) {
        redirectAttributes.addFlashAttribute("testoNotifica",
            "toast.autorizzazioni.richiestaNonAutorizzata");
        return "redirect:/";
    }

    if (utente instanceof ImpiegatoUfficioTirocini) {
        return "redirect:/dashboard/tirocini";
    } else if (utente instanceof DelegatoAziendale) {
        return "redirect:/dashboard/tirocini";
    } else if (utente instanceof Studente) {
        return "redirect:/dashboard/tirocini";
    } else if (utente instanceof PresidenteConsiglioDidattico) {
        return "redirect:/dashboard/tirocini";
    }

    return "redirect:/errore";
}

```

5.3.2 02-CR2:Servizio di Mailing

Per l'implementazione del servizio di mailing è stato semplicemente necessario invocare il metodo `sendEmail(Object obj, String destinatario)` definito in una nuova classe `TirocinioSmartMailSender`, in ciascuno dei metodi definiti nella tabella seguente:

Classe	Metodi
DomandaTirocinioService	<ul style="list-style-type: none">registraDomandaTirocinio(DomandaTirocinio domanda)accettaDomandaTirocinio(long idDomanda, String commentoAzienda)rifiutaDomandaTirocinio(long idDomanda, String commento)approvaDomandaTirocinio(long idDomanda)respingiDomandaTirocinio(long idDomanda, String commento)validaDomandaTirocinio(long idDomanda)annullaDomandaTirocinio(long idDomanda, String commento)
ConvenzioniService	<ul style="list-style-type: none">registraRichiestaConvenzionamento(Azienda azienda)approvaRichiestaConvenzionamento(long idRichiesta)rifiutaRichiestaConvenzionamento(long idRichiesta, String commento)
StudentiService	<ul style="list-style-type: none">registraRichiestalscrizione(Studente studente)approvaRichiestalscrizione(long idRichiesta)rifiutaRichiestalscrizione(long idRichiesta, String commento)

Di seguito è riportata una porzione di codice indicante l'invocazione del metodo sopra descritto. In tal caso, si ha l'invio di una mail quanto la richiesta d'iscrizione di uno studente viene rifiutata dall'Impiegato dell'UfficioTirocini.

```
@Transactional(rollbackFor = Exception.class)
public RichiestaIscrizione rifiutaRichiestaIscrizione(long idRichiesta, String commento)
    throws IdRichiestaIscrizioneNonValidoException, RichiestaIscrizioneGestitaException,
        CommentoRichiestaIscrizioneNonValidoException, RichiestaNonAutorizzataException {
    // Solamente gli impiegati dell'ufficio tirocini possono rifiutare le richieste d'iscrizione
    if (!(utenzaService.getUtenteAutenticato() instanceof ImpiegatoUfficioTirocini)) {
        throw new RichiestaNonAutorizzataException();
    }

    // Controlla che la richiesta esista
    RichiestaIscrizione richiesta = richiestaIscrizioneRepository.findById(idRichiesta);
    if (richiesta == null) {
        throw new IdRichiestaIscrizioneNonValidoException();
    }

    // Controlla che la richiesta non sia già stata gestita in precedenza ed impostare lo stato
    if (richiesta.getStatus() != RichiestaIscrizione.IN_ATTESA) {
        throw new RichiestaIscrizioneGestitaException();
    } else {
        richiesta.setStatus(RichiestaIscrizione.RIFIUTATA);
    }

    richiesta.setCommentoUfficioTirocini(validaCommentoRichiesta(commento));
    mailSender.sendEmail(richiesta, richiesta.getStudente().getEmail());
    return richiesta;
}
```

5.3.3 Actual Impact Set finale

Componente
DomandaTirocinio

DomandeTirocinioService
DomandaTirocinioController
UtenzaService
UtenzaController
StudentiService
ConvenzioniService

5.4 False Positive Impact Set

Componente
DomandaTirocinioRepository
AutenticazioneInterceptor
RichiestaIscrizioneController
ConvenzioniController

5.5 Metriche per la valutazione del processo di Impact Analysis

Nel processo di impact analysis è importante ridurre al minimo le differenze tra AIS e CIS, eliminando i falsi positivi e identificando gli impatti reali. Per verificare quanto è stato utile mettere in campo un processo di analisi degli impatti, andiamo a calcolare alcune metriche.

Due tra queste risultano essere metriche fondamentali ovvero:

- **Recall:** la frazione di impatti reali contenuti nel Candidate Impact Set;
- **Precision:** la frazione di impatti candidati (possibili) che sono effettivamente poi diventati impatti reali.

Per calcolare queste metriche è necessario considerare la cardinalità degli insiemi precedentemente definiti:

- $|SIS| = 5$;
- $|CIS| = 11$;
- $|AIS| = 7$;
- $|DIS| = 0$;
- $|FPIS| = 4$.

$$\text{Recall} = |CIS \cap AIS| / |AIS| = (11 \cap 7) / 7 = 7 / 7 = \mathbf{100\%}$$

$$\text{Precision} = |CIS \cap AIS| / |CIS| = (11 \cap 7) / 11 = 7 / 11 = \mathbf{64\%}$$

Affinchè il processo di impact analysis possa essere ritenuto efficiente entrambe le metriche appena calcolate devono presentare un valore maggiore del 50%.

Ancora, siamo interessati a trarre delle conclusioni relativamente l'adeguatezza del processo in esame, ovvero, vogliamo valutare la capacità dell'approccio di identificare tutti gli elementi coinvolti nella modifica. Ovviamente, per fare ciò calcoliamo la metrica di inclusione:

$$\text{Inclusiveness} = 1 \text{ poiché } \text{AIS} \subseteq \text{CIS}.$$

Dunque, l'inclusiveness ci garantisce che l'approccio può essere utilizzato perché fornisce al manutentore informazioni utili ai fini dell'implementazione delle modifiche.

Inoltre, siamo interessati a definire quanto bene l'approccio si presta a produrre un CIS che sia il più vicino possibile all'AIS. Una piccola differenza tra CIS e AIS ovviamente significa che un piccolo numero di componenti candidate non è interessato dalla modifica. Tale metrica è chiamata **aderenza** e viene calcolata nel seguente modo:

$$|\text{AIS}| / |\text{CIS}| = 7/11 = 0,64.$$

6. Conclusioni

Dalle metriche calcolate nel paragrafo precedente è emerso che sono stati individuati un numero esiguo di falsi positivi, ma comunque, il processo di impact analysis ha riportato una recall ed una precision maggiori del 50%.

Dunque, gli esiti dell'analisi mostrano chiaramente quanto essa sia comunque risultata utile, efficace ed adeguata nel rispondere alle nostre esigenze.

