

Gestione dei dottorati

Relazione a supporto della documentazione

Introduzione

L'obiettivo del progetto consiste nella realizzazione di un applicativo software a microservizi per la gestione dei dottorati dell'Università degli Studi di Firenze.

Si fa riferimento all'organizzazione del dottorato in Smart Computing dell'Università degli Studi di Firenze (<https://smartcomputing.unifi.it/>), il quale viene preso come esempio nella modellazione.

La trattazione che segue cerca di generalizzare il più possibile il contesto applicativo nonostante la possibilità di avere dottorati all'interno della stessa università con struttura e organizzazione diversa in relazione al caso specifico.

Descrizione del contesto

Il corso di dottorato dell'Università degli Studi di Firenze viene offerto agli studenti secondo dei **cicli**, intesi come le annualità dello stesso. Ad ogni ciclo, di conseguenza, afferiscono un insieme di docenti denominati **faculty member** e degli **studenti**. Ad ogni studente è associato un **advisor**, ossia un professore facente parte dei faculty. Al ciclo è associato anche un insieme di **corsi**, i quali possono essere tenuti da **docenti** "generici", ossia possono essere sia docenti esterni che faculty member. Ogni corso sarà fruibile mediante un insieme di **lezioni** programmate ed inserite direttamente dal docente. Lo studente **partecipa** al corso, il quale si conclude di norma con un **esame** che ha vari **appelli** (programmati ed inseriti anch'essi dal docente). Una volta superato, il voto verrà registrato sul **libretto**.

Alla fine di ciascun ciclo, gli studenti del I e II anno dovranno redigere e discutere un **progress report**, ossia una relazione che descrive i risultati scientifici conseguiti nel ciclo appena concluso. Esso verrà revisionato da una commissione di **scienziati** esterni, denominata **supervisory committee**, che dovrà formarsi almeno tre mesi prima del termine di presentazione del progress report.

La commissione ha il compito di esprimere un giudizio in merito al passaggio dello studente all'anno successivo sulla base di una discussione del progress report. In caso negativo lo studente non potrà accedere all'anno successivo.

Per quanto riguarda gli studenti del III anno dovranno caricare sull'applicazione la propria **tesi**. Essa verrà revisionata, mediante l'inserimento di **revisioni** da una commissione di **reviewer** (docenti di elevata qualificazione), denominata **evaluation committee**.

I due valutatori esprimono un giudizio analitico scritto sulla tesi e ne propongono l'ammissione alla discussione pubblica o il rinvio per un periodo non superiore a sei mesi se ritengono necessarie significative integrazioni o correzioni.

L'applicativo, inoltre, presenta una categoria per gli ex-studenti, ossia l'insieme degli studenti che hanno conseguito il titolo di dottore di ricerca. I quali potranno, ad esempio, vedere lo storico degli esami conseguiti nel corso formativo concluso e figurare a titolo informativo insieme al loro lavoro di ricerca.

Requisiti funzionali

I requisiti funzionali catturano il comportamento atteso del sistema in termini di servizi, compiti e funzioni.

Una pratica di successo e dominante nella rappresentazione di essi è l'utilizzo dei casi d'uso, ovvero un insieme di interazioni tra il sistema ed uno o più attori esterni al sistema stesso finalizzate ad un obiettivo.

Per quanto riguarda l'applicativo di nostro interesse abbiamo individuato 5 attori:

- Amministratore (Admin)
- Docente (Professor)
- Studente (Student)
- Scienziato (Scientist)
- Relatore (Reviewer)
- Faculty member

Use Case Diagram

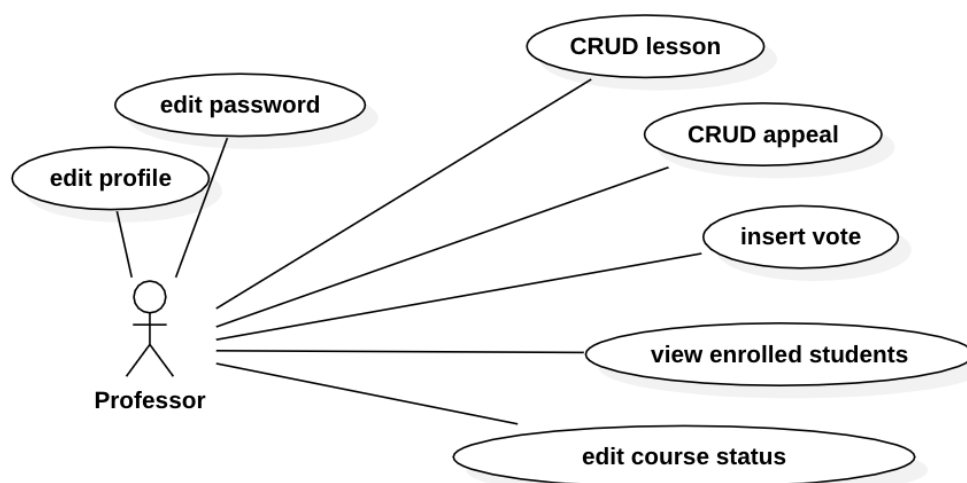
Personale amministrativo

La figura dell'amministratore descrive l'unica figura in primo luogo esterna al dottorato (ad esempio un membro della segreteria della facoltà di appartenenza del dottorato) e responsabile della gestione delle **operazioni basilari (creazioni ed eventuali assegnazioni)** per il funzionamento dell'applicativo.



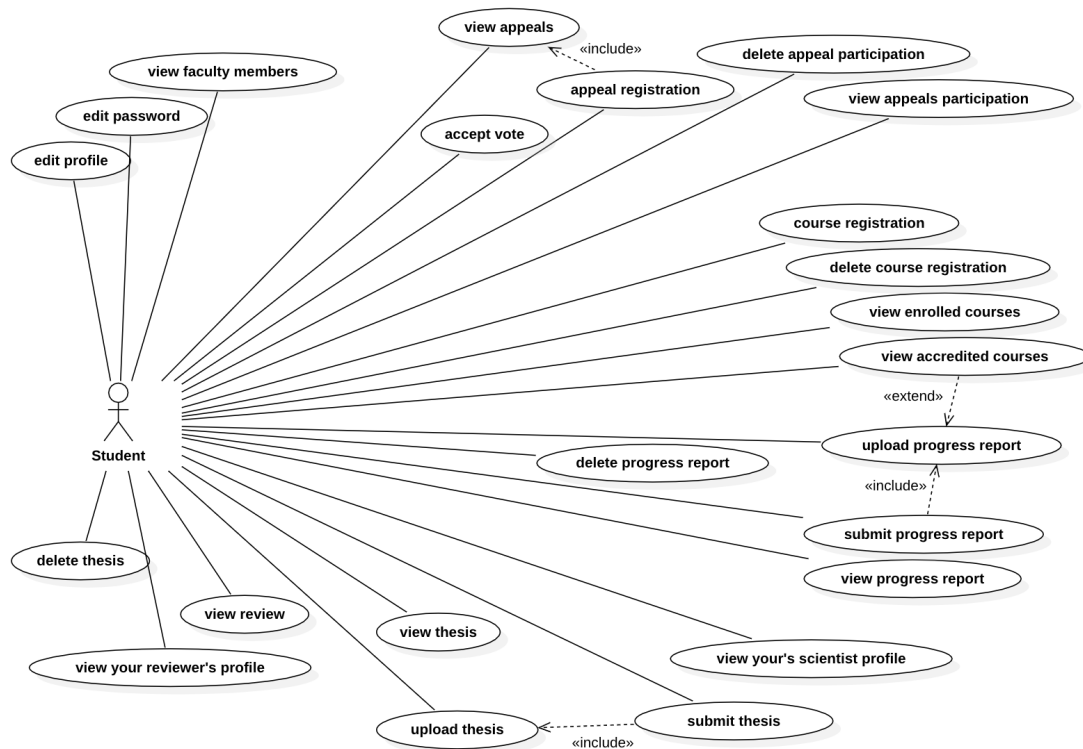
Docente del corso

Il docente interagisce con l'applicativo andando ad agire sul corso in termini di **aggiunta delle lezioni e degli appelli d'esame**. In seguito al superamento dell'esame da parte degli studenti si occupa della **registrazione del voto sul libretto**.



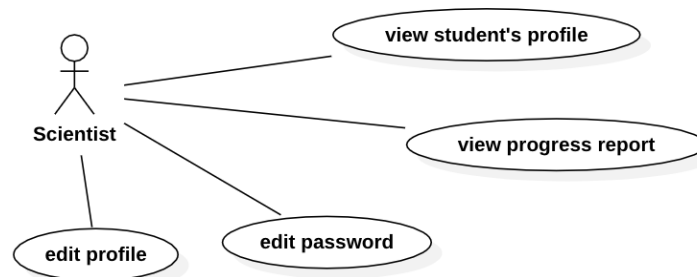
Studente

Lo studente ha la possibilità di **guardare i corsi disponibili** per scegliere a quale **iscriversi**. Per ogni corso che segue può **visualizzarne gli appelli ed iscriversi a questi ultimi**. Può quindi **vedere i corsi superati** mediante la visione del libretto. Inoltre, a seconda che sia al I e II anno o III anno può **redigere il progress report** o **caricare la tesi** (come descritto in precedenza).



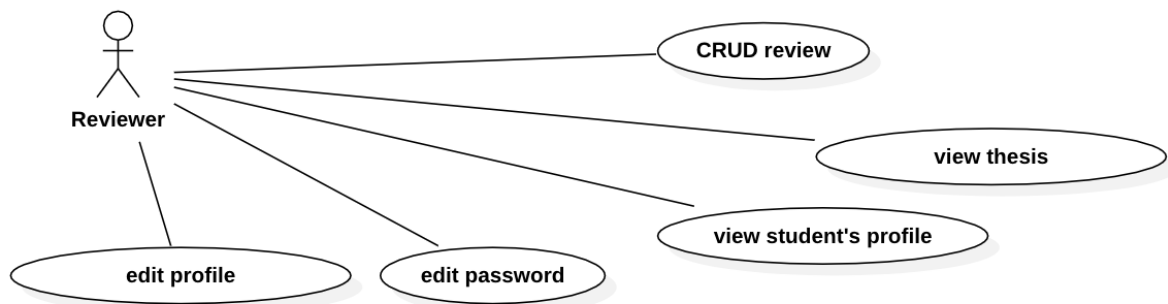
Supervisory Committee

L'interazione tra progress report e Supervisory Committee è stata rappresentata dall'attore Scientist, il quale potrà **visualizzare il progress report**.



Evaluation Committee

L'interazione tra la tesi ed Evaluation Committee è stata rappresentata dall'attore Reviewer. In questo caso oltre a **visionare la tesi dello studente**, potrà **generare delle revisioni** relative ad essa.



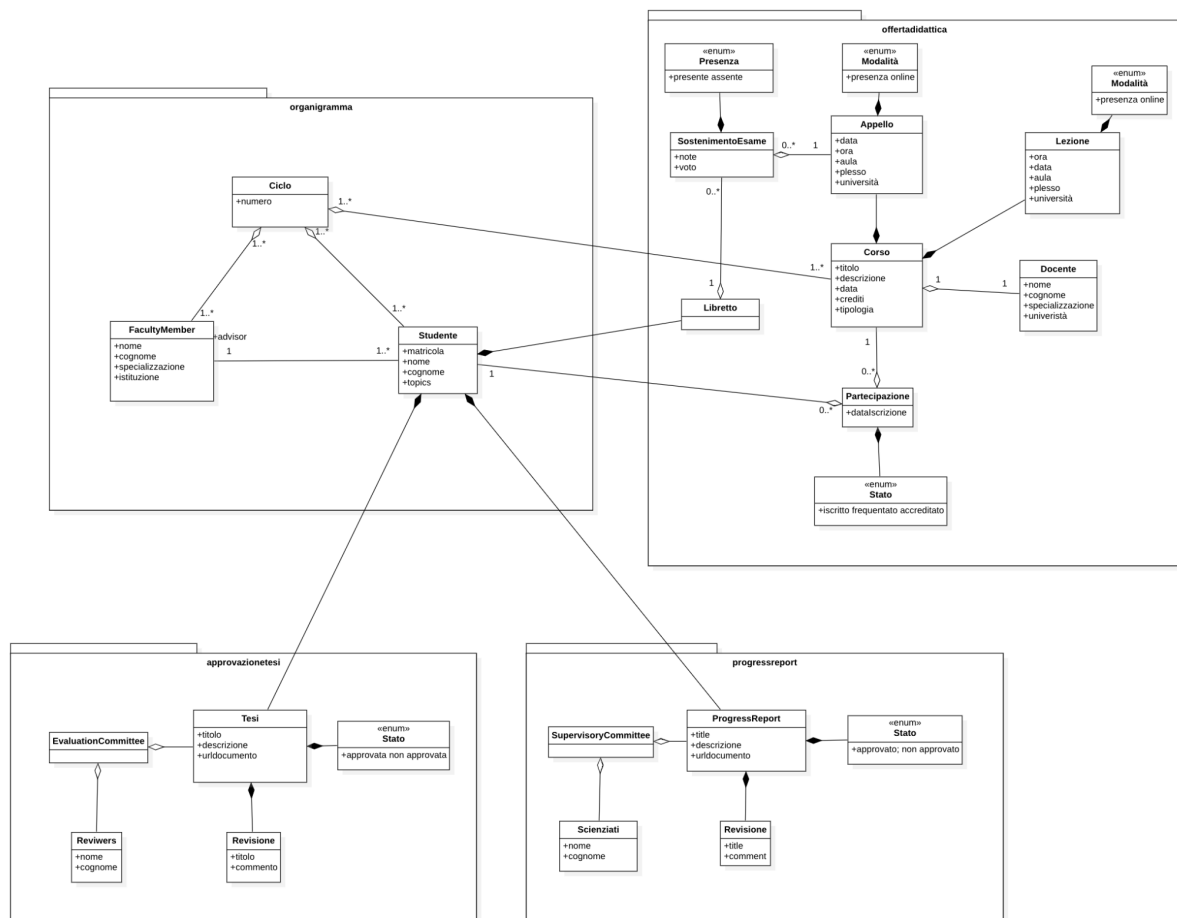
Requisiti di dominio

Dopo aver descritto i requisiti funzionali, ovvero dopo aver definito il ruolo degli utenti esterni e le funzioni del sistema attraverso l'utilizzo dei casi d'uso, è necessario passare alla caratterizzazione dei domini concettuali in termini di classi, oggetti, relazioni e package. Il metodo più semplice per realizzare ciò è utilizzare il linguaggio UML.

Nell'applicativo abbiamo individuato 4 package differenti, ognuno dei quali rappresenterà un microservizio autonomo:

- organigramma (organizationchart)
- offertadidattica (didacticoffer)
- approvazionesi (thesisapproval)
- progressreport

Class Diagram



Organigramma

Cosa rappresenta:

- Rappresenta il **ciclo** del corso
- Rappresenta il **faculty member**
- Rappresenta lo **studente**

Relazioni:

- Ad ogni studente è associato un solo faculty member, denominato **advisor** (in casi particolari, come descritto nel sito di smart computing potrebbero essere associati anche 2, ma al momento non tocchiamo questo livello di dettaglio)
- Ogni ciclo ha associato un insieme di studenti, un insieme di faculty member. Queste relazioni sono state realizzate mediante due aggregazioni.

Offerta didattica

Cosa rappresenta:

- Rappresenta il **corso**
- Rappresenta il **docente** (il quale non per forza deve far parte dei faculty member)
- Rappresenta gli **appelli** del corso
- Rappresenta il **libretto** dello studente
- Rappresenta la **lezione** (relativa ad un corso)

Relazioni:

- La relazione tra lezione e corso è rappresentata da una composizione, stesso discorso per la relazione tra appello e corso
- Uno studente può partecipare a più corsi ed ad un corso possono partecipare più studenti, la relazione tra studente e corso è esplicitata mediante un mapper di **partecipazione**
- Uno studente può sostenere più appelli e uno stesso appello può essere sostenuto da più studenti, la relazione tra studente ed appelli è esplicitata mediante un mapper di **esame sostenuto**
- Un corso può essere tenuto da un solo docente ed un docente può tenere un solo corso.

Progress report

Cosa rappresenta:

- Rappresenta il **progress report**
- Rappresenta la **supervisory committee**
- Rappresenta lo **scienziato**

Relazioni:

- La relazione tra progress report e scienziati è gestita mediante l'utilizzo di un mapper denominato come **supervisory committee**

Approvazione tesi

Cosa rappresenta:

- Rappresenta la **tesi**
- Rappresenta l'**evaluation committee**
- Rappresenta la **revisione**
- Rappresenta il **relatore**

Relazioni:

- Inoltre, alla tesi è associata, mediante l'utilizzo di un'aggregazione, l'evaluation committee, la quale ha la funzione di mappere tra tesi e il reviewer.
- L'evaluation committee sarà composta da due relatori e questa relazione è stata rappresentata mediante l'utilizzo di un'aggregazione.
- La tesi avrà associato una o più revisioni, redatte da uno specifico relatore. Questa relazione è stata rappresentata mediante l'utilizzo di una composizione in quanto una revisione non avrebbe significato senza l'esistenza della tesi.

Package Diagram

La rappresentazione del package diagram si è resa necessaria per esplicitare la necessità dei microservizi (packages) di collaborare tra loro.

Ogni package è stato schematizzato come segue:

- Package che rappresenta il singolo microservizio (offerta didattica, organigramma, ...), il cui **domain model** viene omissso in quanto già presente nel capitolo precedente del class diagram
- Subpackage **server** dentro il quale vengono rappresentate le classi utili ai fini di rendere a disposizione le funzionalità dei casi d'uso e di responsabilità del micro servizio in oggetto (livello business logic)
- Subpackage **proxy** dentro il quale vengono rappresentate le classi per la gestione della comunicazione tra i vari microservizi
- Subpackage di **data access** dentro il quale vengono rappresentate le classi utili ai fini di implementare la comunicazione con lo strato di persistenza
- **Database locale** (non rappresentato graficamente), il quale contiene i dati memorizzati dal singolo microservizio

