

POLITECNICO DI MILANO  
Facoltà di Ingegneria dell'Informazione  
Corso di Laurea in Ingegneria Informatica



## Progetto di Ingegneria del Software 2

### Parte I: RASD con rettifiche

Autori:

**Paolo FERRARIS (matr. 716032)**

p.ferraris88@gmail.com

**Fabio MONTI (matr. 782577)**

f.monti88@teletu.it

**Elisabetta A. MORELLI (matr. 782557)**

elisabetta.morelli@libero.it

Prof.ssa: Di Nitto Elisabetta

**Anno accademico 2011/12**

# INDICE

Elenco delle figure .....	4
Elenco delle tabelle .....	5
Descrizione del progetto .....	7
1. Introduzione .....	9
1.1 Obiettivi generali.....	9
1.2 Glossario.....	10
1.3 Considerazioni preliminari .....	11
2. Dominio.....	12
2.1 Entità.....	12
2.2 Stakeholders.....	13
2.3 Relazioni.....	13
3. Requisiti.....	14
3.1 Requisiti funzionali.....	14
3.2 Requisiti non funzionali.....	15
3.2.1 Interfacce utente e fattore umano.....	15
3.2.2 Documentazione .....	15
3.2.3 Architettura del sistema.....	16
3.2.4 Hardware .....	16
3.2.5 Gestione degli errori e condizioni estreme .....	16
3.2.6 Sicurezza .....	16

4. Modelli.....	17
4.1 Scenari.....	17
4.1.1 Studente.....	17
4.1.2 Professore .....	19
4.1.3 Amministratore .....	20
4.2 Casi d'uso .....	20
4.2.1 Utente.....	21
4.2.2 Studente.....	22
4.2.3 Professore .....	26
4.2.4 Amministratore .....	29
4.3 Diagrammi .....	30
4.3.1 Class Diagram .....	31
4.3.2 Sequence Diagram .....	32
4.3.3 Activity Diagram.....	36
5. Specifiche.....	40
5.1 Definizione delle specifiche.....	39
5.2 Verifica delle specifiche.....	40
5.2.1 Signatures.....	40
5.2.2 Facts .....	41
5.2.3 Assertions & Predicates .....	43
5.2.4 Mondo generato .....	45
Appendice A. Software utilizzati .....	48
A.1 StarUML.....	48
A.2 Alloy Analyzer 4.0 .....	48

## ELENCO DELLE FIGURE

Fig. 4.1 – Use Case Diagram .....	21
Fig. 4.2 – Use Case Diagram: Studente .....	22
Fig. 4.3 – Use Case Diagram: Professore.....	26
Fig. 4.4 – Use Case Diagram: Amministratore.....	29
Fig. 4.5 – Class Diagram.....	31
Fig. 4.6 – Sequence Diagram: Registrazione studente .....	32
Fig. 4.7 – Sequence Diagram: Creazione nuovo gruppo .....	32
Fig. 4.8 – Sequence Diagram: Upload release .....	33
Fig. 4.9 – Sequence Diagram: Valutazione release .....	34
Fig. 4.10 – Sequence Diagram: Condividi release .....	35
Fig. 4.11 – Activity Diagram: Registrazione studente .....	36
Fig. 4.12 – Activity Diagram: Gestione release .....	37
Fig. 4.13 – Activity Diagram: Valutazione release .....	38
Fig. 4.14 – Activity Diagram: Pubblica progetto .....	38
Fig. 5.1 – Mondo generato Alloy.....	47

## ELENCO DELLE TABELLE

Tab. 1.1 – Glossario .....	10
Tab. 4.1 – Scenario: Registrazione studente .....	17
Tab. 4.2 – Scenario: Formazione gruppo .....	18
Tab. 4.3 – Scenario: Upload release .....	18
Tab. 4.4 – Scenario: Valutazione release .....	19
Tab. 4.5 – Scenario: Condivisione release .....	19
Tab. 4.6 – Scenario: Pubblicazione progetto.....	20
Tab. 4.7 – Scenario: Registrazione professore .....	20
Tab. 4.8 – Caso d’uso: Login.....	21
Tab. 4.9 – Caso d’uso: Registrazione .....	22
Tab. 4.10 – Caso d’uso: Creazione nuovo gruppo.....	22
Tab. 4.11 – Caso d’uso: Unione gruppo disponibile .....	23
Tab. 4.12 – Caso d’uso: Modifica profilo .....	23
Tab. 4.13 – Caso d’uso: Upload release al progetto .....	23
Tab. 4.14 – Caso d’uso: Visualizza release del progetto.....	24
Tab. 4.15 – Caso d’uso: Visualizza voto.....	24
Tab. 4.16 – Caso d’uso: Visualizza release condivise.....	25
Tab. 4.17 – Caso d’uso: Download release .....	25
Tab. 4.18 – Caso d’uso: Pubblica progetto.....	26
Tab. 4.19 – Caso d’uso: Elenco release per tipo .....	26

Tab. 4.20 – Caso d’uso: Elenco release per gruppo .....	27
Tab. 4.21 – Caso d’uso: Condividi release .....	28
Tab. 4.22 – Caso d’uso: Valuta release .....	28
Tab. 4.23 – Caso d’uso: Download release .....	28
Tab. 4.24 – Caso d’uso: Registra professore .....	29
Tab. 4.25 – Caso d’uso: Visualizza profilo professore .....	30
Tab. 4.26 – Caso d’uso: Elimina professore .....	30

## DESCRIZIONE DEL PROGETTO

**M**anage Project Homework (MPH) è un'applicazione usata congiuntamente da professori e studenti per facilitare l'organizzazione dei corsi universitari per la consegna dei progetti.

Gli studenti lavorano insieme in gruppi composti da 1, 2 o 3 partecipanti e accedono al sistema attraverso username e password. Ogni studente, dopo essersi registrato, possiede un profilo standard contenente le sue informazioni personali (nome, cognome, matricola, foto, ecc.).

I professori, la cui registrazione viene effettuata dall'amministratore, utilizzano il sistema per pubblicare la descrizione del progetto e per definire le diverse consegne, caratterizzate da un nome e una specifica deadline (ad esempio "Delivery of Design Document" da consegnare entro il giorno 15 Novembre 2011 alle ore 23.59).

Ogni professore può fare il download delle consegne in due modalità:

- in base al gruppo, ad esempio è possibile fare il download di tutte le consegne prodotte da un dato gruppo;
- in base al tipo di consegna, ad esempio è possibile fare il download di tutti i RASD presenti sul sistema.

I professori, attraverso un appropriato input form, possono valutare le consegne di ogni gruppo con un punteggio compreso tra 1 e 10 (ai documenti non consegnati verrà automaticamente assegnato il punteggio 1, mentre ai documenti consegnati in ritardo verrà corrisposta una penalità proporzionale al ritardo di consegna). Gli

studenti possono visualizzare il voto relativo al documento consegnato solo dopo che il professore lo ha pubblicato sul sistema.

Il sistema provvederà automaticamente ad assegnare un voto finale al gruppo, ottenuto come media di tutti i voti intermedi sulle singole consegne.

Infine il professore può utilizzare il sistema per condividere materiale tra i gruppi; in particolare egli seleziona il gruppo che deve avere visibilità sui documenti prodotti da altri gruppi. Naturalmente, questo si traduce nel fatto che il primo gruppo avrà a disposizione alcune funzionalità in più che gli permettono l'accesso al materiale condiviso.





# INTRODUZIONE

**I**n questo capitolo verranno presentati gli obiettivi generali del progetto, il contesto in cui esso si è sviluppato e le assunzioni che gli autori hanno fatto nel corso della progettazione. Per una migliore comprensione del documento è stato introdotto un mini glossario nel quale verranno chiariti i significati di determinate parole o sigle utilizzate.

## 1.1 Obiettivi generali

L'obiettivo principale di MPH è la creazione di un sistema informatico di supporto nella consegna dei progetti da parte di studenti di un determinato corso di studi.

Il sistema dovrà fornire le seguenti funzionalità:

- permettere ai professori di pubblicare la **descrizione del progetto**;
- permettere ai professori di definire le diverse consegne, caratterizzate da un nome e una specifica deadline;
- permettere ai professori di effettuare il download delle release;
- permettere ai professori di valutare le release prodotte da ogni gruppo e di pubblicare le valutazioni sul sistema;
- permettere ai professori di condividere materiale tra i gruppi;
- permettere agli studenti di registrarsi al sistema e di formare i gruppi;

- permettere agli studenti di fare l'upload delle release;
- permettere agli studenti di visualizzare la valutazione parziale delle singole release e la valutazione finale del progetto;
- permettere agli studenti di visualizzare il materiale condiviso.

## 1.2 Glossario

Per una corretta comprensione dei termini utilizzati nel presente documento, si forniscono le seguenti definizioni:

Materiale	È il materiale che il professore del corso pubblica sul portale ed è visibile a tutti i gruppi. In particolare contiene la descrizione del progetto e la definizione delle deadline.
Release	Rappresentano i files di consegna che il gruppo carica sul sistema, i quali verranno in seguito valutati dal professore.
Deadline	È una specifica data per la quale lo studente deve consegnare la release.
Penalità	Rappresenta un valore numerico che andrà a sottrarsi al voto proposto dal professore, per coloro che hanno fatto l'upload della release successivamente alla deadline. Essa viene attribuita automaticamente dal sistema in seguito all'upload della release stessa.
Voto parziale	È la valutazione che viene assegnata ad una data release; viene calcolata come sottrazione tra il voto presentato dal professore e la penalità attribuita dal sistema.
Voto finale	È la valutazione complessiva del progetto; viene calcolata come media dei voti parziali associati alle release.
File sharing	Letteralmente significa "condivisione di file"; in questo contesto permette al professore di condividere una o più release tra due o più gruppi.
Progetto	Rappresenta il lavoro svolto dai singoli gruppi sulla base del materiale pubblicato dal professore.

Tabella 1.1 – Glossario

### 1.3 Considerazioni preliminari

In questo paragrafo verranno discussi gli aspetti poco chiari riscontrati nella descrizione del problema per i quali gli autori di questo progetto hanno fatto le seguenti assunzioni, adeguatamente motivate:

- *Modalità di calcolo del voto parziale su una data release*: nel momento in cui lo studente fa l'upload della release, il sistema automaticamente registra la data in cui è stata effettuata tale operazione. Confrontando la data di upload con la deadline definita dal professore, il sistema verifica se il documento è stato consegnato per tempo. Nel caso in cui lo studente non abbia rispettato le scadenze prefissate, il sistema calcola una penalità sulla base del numero dei giorni di ritardo.
- *Visibilità dei profili*: i professori possono visualizzare solo il proprio profilo mentre gli studenti possono visualizzare il proprio profilo e il profilo dei gruppi a cui appartengono. L'amministratore del sistema, invece, può visualizzare il profilo dei professori, degli studenti e di tutti i gruppi presenti nel sistema.



## DOMINIO

**P**er dominio si intende il mondo reale in cui il sistema andrà ad inserirsi ed interfacciarsi. A tal proposito verranno di seguito definite le entità di interesse, gli stakeholders e le relazioni.

### 2.1 Entità

Dall'analisi del problema si sono individuate le seguenti entità-chiave:

- Utente
- Studente
- Professore
- Amministratore
- Gruppo
- Progetto
- Release

## 2.2 Stakeholders

Le principali persone alle quali il progetto si rivolge sono:

- Gli studenti iscritti al corso, che hanno come obiettivo quello di terminare il progetto, presentando ogni documento entro le scadenze prefissate dal professore;
- Il professore del corso, che ha come obiettivo quello di assegnare un progetto agli studenti e di valutarlo in tutte le sue parti;
- L'amministratore del sistema, che ha come obiettivo quello di registrare i professori e rendere la piattaforma il più possibile efficiente e funzionale in relazione agli obiettivi esposti nell'introduzione.

## 2.3 Relazioni

Durante la stesura del progetto sono state individuate le seguenti relazioni tra entità:

- Studente, Professore e Amministratore sono Utenti del sistema.
- Ogni Gruppo è formato da 1, 2 o 3 Studenti.
- Ogni Studente appartiene a uno o più Gruppi.
- Ogni Professore è responsabile di uno o più Progetti.
- Ogni Progetto è seguito da un solo Professore.
- Ogni Progetto viene svolto da un solo Gruppo.
- Ogni Gruppo svolge uno o più Progetti.
- Ogni Progetto è suddiviso in una o più Release.
- Ogni Release fa riferimento ad un solo Progetto.
- Ogni Release può essere visibile a uno o più Gruppi (oltre al Gruppo proprietario).
- Ogni Gruppo può visualizzare una o più Release (oltre le Release da lui pubblicate).



## REQUISITI

Una delle prime fasi del ciclo di vita di un prodotto software è l'analisi dei requisiti, il cui scopo principale è quello di stabilire che cosa il sistema globalmente deve fare. È necessario capire le esigenze che determinano la necessità di un nuovo sistema e analizzare lo scenario del cliente, ponendo particolare attenzione alle risorse disponibili.

### 3.1 Requisiti funzionali

I requisiti funzionali hanno il compito di descrivere le funzioni e i servizi del sistema. In questo paragrafo verranno esposti i requisiti funzionali, sulla base delle funzionalità emerse per ogni attore.

- Le funzionalità richieste per lo Studente comprendono:
  - La registrazione nel sistema, quindi l'inserimento dei dati personali e la successiva creazione di un gruppo.
  - La gestione delle proprie release, quindi l'upload delle stesse e la visualizzazione delle valutazioni ad esse associate.
  - La visualizzazione delle release di altri gruppi, qualora il professore ne abiliti la condivisione.

- Le funzionalità richieste per il Professore comprendono:
  - La creazione di un nuovo progetto, quindi l’inserimento del nome, della descrizione e la definizione delle deadline.
  - La valutazione delle release.
  - La condivisione delle release tra i gruppi.
- Le funzionalità richieste per l’Amministratore comprendono:
  - La registrazione dei professori nel portale.
  - La visualizzazione dei profili di tutti gli utenti del sistema, compresa l’eventuale eliminazione degli stessi dal database.

## **3.2 Requisiti non funzionali**

I requisiti non funzionali non sono strettamente inerenti al comportamento funzionale del sistema, ma essi risultano comunque utili per descrivere gli aspetti di interesse per gli stakeholders.

### **3.2.1 Interfacce utente e fattore umano**

Il sistema utilizza interfacce grafiche di tipo object-oriented basate sul web, la cui caratteristica principale è data dall’usabilità, la quale consente al cliente un facile utilizzo della piattaforma. La struttura grafica, la quale si differenzia in base al ruolo dell’utente, permette un’immediata individuazione delle funzionalità principali che il sistema offre.

### **3.2.2 Documentazione**

Allo scopo di illustrare lo sviluppo del progetto verranno realizzati i seguenti documenti:

- Requirements Analysis Specification Document (RASD), rivolto al cliente, agli utenti finali e agli sviluppatori;
- Design Document (DD), rivolto agli sviluppatori;
- Test cases;

- Manuale d'uso, rivolto agli utenti;
- Documento di test.

### **3.2.3 Architettura del sistema**

Il sistema verrà sviluppato mediante la piattaforma JEE, pertanto sarà necessario che su tutti i terminali venga installata una delle ultime versioni della Java Virtual Machine e un Web Browser per l'interfacciamento con l'utente.

### **3.2.4 Hardware**

Considerato il tipo di architettura previsto e il requisito di persistenza dei dati, si prevede la necessità di un server collegato tramite rete WAN ai terminali degli utenti. Tale server dovrà adottare soluzioni, quali impianto di continuità, in grado di salvaguardare il sistema dalla perdita dei dati, anche in caso di mancanza di corrente.

### **3.2.5 Gestione degli errori e condizioni estreme**

Il sistema deve poter garantire la coerenza e la persistenza dei dati immagazzinati, anche in caso di perdita di connessione ed errori di upload o di aggiornamento.

### **3.2.6 Sicurezza**

Poiché ogni utente accede alle sole funzionalità che gli competono, è necessario prevedere un meccanismo di autenticazione (login) con username e password a cui si accede all'avvio del programma.





## MODELLI

**I**l presente capitolo è dedicato all'illustrazione dei concetti base forniti dall'UML (Unified Modeling Language) per supportare il delicato processo di analisi dei requisiti da parte dell'utente. In particolare, si porrà attenzione alla descrizione degli scenari e alla modellizzazione dei casi d'uso cercando di astrarsi il più possibile dall'implementazione, ossia concentrarsi maggiormente su cosa il sistema dovrà fare piuttosto che su come lo dovrà fare. Infine verranno riportati i diagrammi atti a descrivere i principali casi d'uso rilevati in fase di progettazione.

### 4.1 Scenari

Per scenario si intende una descrizione narrativa dell'esperienza che gli attori fanno all'interno del sistema. In seguito verranno presentati i principali scenari, raggruppati per attore, emersi durante la modellizzazione del problema.

#### 4.1.1 Studente

Nome dello scenario	Registrazione studente
Attori partecipanti	Ambra: Studente
Flusso degli eventi	1. Ambra accede al sistema attraverso il Browser.

	<ol style="list-style-type: none"> <li>2. Il sistema presenta l'interfaccia Registrazione.</li> <li>3. Ambra compila le varie form di registrazione.</li> <li>4. Il sistema verifica i dati immessi e registra lo studente.</li> </ol>
--	--

Tabella 4.1 – Scenario: Registrazione studente

Nome dello scenario	Formazione gruppo
Attori partecipanti	Ambra: Studente
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Ambra accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Studente.</li> <li>3. Ambra avvia la procedura di Formazione gruppo.</li> <li>4. Il sistema propone l'interfaccia per scegliere un team di lavoro.</li> <li>5. Ambra decide di creare un nuovo gruppo.</li> <li>6. Il sistema notifica l'avvenuta creazione del gruppo e chiede di scegliere un progetto da svolgere.</li> <li>7. Ambra sceglie il progetto tra quelli disponibili.</li> <li>8. Il sistema notifica l'avvenuta operazione.</li> </ol>

Tabella 4.2 – Scenario: Formazione gruppo

Nome dello scenario	Upload release
Attori partecipanti	Ambra: Studente
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Ambra accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Studente.</li> <li>3. Ambra accede all'area Elenco progetti.</li> <li>4. Il sistema risponde visualizzando l'elenco dei progetti su cui Ambra sta lavorando.</li> <li>5. Ambra seleziona il progetto MPH e avvia la procedura di Upload release.</li> <li>6. Il sistema propone l'interfaccia per effettuare l'upload delle release.</li> <li>7. Ambra sceglie di fare l'upload della Release DD e avvia l'upload.</li> <li>8. Il sistema verifica che la data di upload è successiva alla deadline fissata e calcola la penalità.</li> <li>9. Il sistema notifica allo studente la presenza di una penalità e l'esito positivo dell'upload.</li> </ol>

Tabella 4.3 – Scenario: Upload release

#### 4.1.2 Professore

Nome dello scenario	Valutazione release
Attori partecipanti	Riccardo: Professore
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Riccardo accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Professore.</li> <li>3. Riccardo accede all'area Elenco progetti.</li> <li>4. Il sistema risponde visualizzando l'elenco dei progetti che Riccardo sta seguendo.</li> <li>5. Riccardo sceglie di valutare il progetto MPH.</li> <li>6. Il sistema restituisce l'interfaccia riassuntiva del progetto MPH.</li> <li>7. Riccardo decide di valutare i documenti RASD.</li> <li>8. Il sistema propone l'interfaccia contenente l'elenco dei RASD consegnati.</li> <li>9. Riccardo seleziona il RASD del Gruppo 1 e ne effettua il download.</li> <li>10. Riccardo accede alla form di valutazione e inserisce il voto nel sistema.</li> <li>11. Il sistema verifica la presenza di eventuali penalità del Gruppo 1 e calcola il voto finale del RASD.</li> <li>12. Il sistema notifica il voto calcolato.</li> </ol>

Tabella 4.4 – Scenario: Valutazione release

Nome dello scenario	Condivisione release
Attori partecipanti	Riccardo: Professore
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Riccardo accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Professore.</li> <li>3. Riccardo accede all'area Elenco progetti.</li> <li>4. Il sistema risponde visualizzando l'elenco dei progetti che Riccardo sta seguendo.</li> <li>5. Riccardo accede al progetto MPH.</li> <li>6. Il sistema restituisce l'interfaccia riassuntiva del progetto MPH.</li> <li>7. Riccardo decide di rendere visibile la Release PLAN al Gruppo 2.</li> <li>8. Il sistema notifica l'avvenuta condivisione della release.</li> </ol>

Tabella 4.5 – Scenario: Condivisione release

Nome dello scenario	Pubblicazione progetto
Attori partecipanti	Riccardo: Professore
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Riccardo accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Professore.</li> <li>3. Riccardo avvia la procedura Pubblica progetto.</li> <li>4. Il sistema propone l'interfaccia per la creazione di un nuovo progetto.</li> <li>5. Riccardo inserisce il nome del progetto, la descrizione e inserisce tramite opportune form le deadline.</li> <li>6. Il sistema notifica l'esito positivo dell'operazione.</li> </ol>

Tabella 4.6 – Scenario: Pubblicazione progetto

### 4.1.3 Amministratore

Nome dello scenario	Registrazione professore
Attori partecipanti	Steve: Amministratore
Flusso degli eventi	<ol style="list-style-type: none"> <li>1. Steve accede al portale inserendo username e password.</li> <li>2. Il sistema propone l'interfaccia Amministratore.</li> <li>3. Steve avvia la procedura Registrazione professore.</li> <li>4. Il sistema propone l'interfaccia per registrare un nuovo professore.</li> <li>5. Steve inserisce i dati del professore nelle opportune form e avvia la registrazione.</li> <li>6. Il sistema notifica l'esito positivo dell'operazione.</li> </ol>

Tabella 4.7 – Scenario: Registrazione professore

## 4.2 Casi d'uso

I casi d'uso rappresentano le principali funzionalità richieste dagli stakeholders; in particolari i Diagrammi dei Casi d'Uso (Use Case Diagram) definiscono il comportamento del sistema dal punto di vista dell'utente. In seguito verranno presentati i principali casi d'uso, raggruppati per attore, derivanti dagli scenari esposti nel paragrafo 4.1.

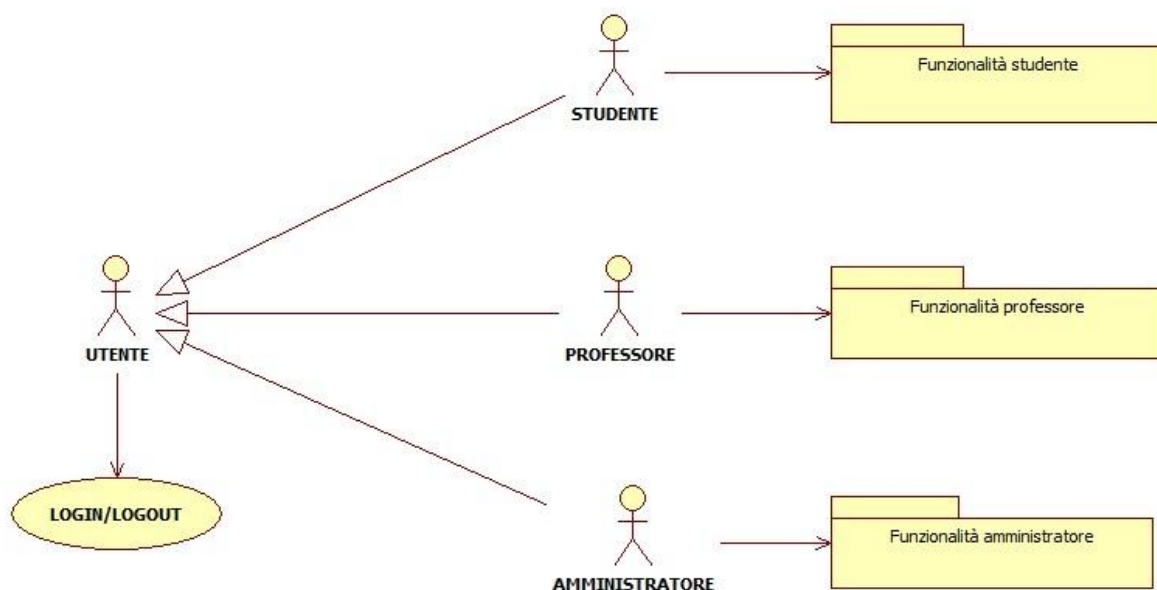


Figura 4.1 – Use Case Diagram

## 4.2.1 Utente

Nome del caso d'uso	Login
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia MPH.
Flusso degli eventi	2. MPH risponde visualizzando la form di inserimento di username e password per l'autenticazione. 3. L'Utente inserisce i propri dati e conferma.
Condizione d'uscita	4. MPH verifica l'autenticazione dell'Utente e visualizza l'interfaccia con le funzionalità relative al tipo di accesso.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 4, MPH verifica che lo username e/o la password inserita è errata, lo comunica all'Utente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.8 – Caso d'uso: Login

## 4.2.2 Studente

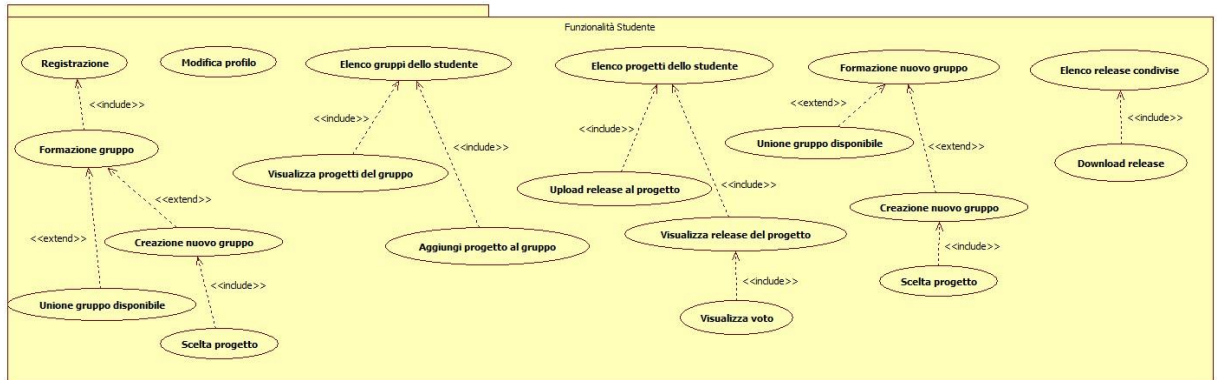


Figura 4.2 – Use Case Diagram: Studente

Nome del caso d'uso	Registrazione
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia MPH.
Flusso degli eventi	2. MPH risponde visualizzando la form di inserimento dei dati personali. 3. L'Utente inserisce i propri dati e conferma.
Condizione d'uscita	4. MPH verifica che lo username scelto dallo Studente non sia già in uso e conferma l'avvenuta registrazione.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH verifica che lo username scelto è già in uso, lo comunica all'Utente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.9 – Caso d'uso: Registrazione

Nome del caso d'uso	Creazione nuovo gruppo
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per formare un team di lavoro.
Flusso degli eventi	2. MPH risponde visualizzando l'interfaccia per la formazione di un team di lavoro. 3. Lo Studente decide di creare un nuovo gruppo. 4. MPH risponde visualizzando l'interfaccia per la scelta del progetto. 5. Lo Studente sceglie il progetto e conferma.
Condizione d'uscita	6. MPH verifica la disponibilità del nome proposto

	per il nuovo gruppo.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 6, MPH verifica che il nome è già stato utilizzato, lo comunica allo Studente e il caso d'uso riprende dal punto 3.</li> </ul>

Tabella 4.10 – Caso d'uso: Creazione nuovo gruppo

Nome del caso d'uso	Unione gruppo disponibile
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per formare un team di lavoro.
Flusso degli eventi	2. MPH risponde visualizzando l'interfaccia per la formazione di un team di lavoro. 3. Lo Studente decide di unirsi ad un gruppo.
Condizione d'uscita	4. MPH notifica allo Studente che l'operazione si è conclusa con successo.
Eccezioni	-----

Tabella 4.11 – Caso d'uso: Unione gruppo disponibile

Nome del caso d'uso	Modifica profilo
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per modificare i dati personali.
Flusso degli eventi	2. MPH risponde visualizzando l'interfaccia per la modifica dei dati personali. 3. Lo Studente modifica i dati di suo interesse e conferma.
Condizione d'uscita	4. MPH verifica che i dati inseriti siano coerenti e notifica che l'operazione si è conclusa con successo.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 4, MPH verifica che i dati inseriti non sono coerenti, lo comunica allo Studente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.12 – Caso d'uso: Modifica profilo

Nome del caso d'uso	Upload release al progetto
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per accedere al progetto nel quale

	caricare la release.
Flusso degli eventi	<ol style="list-style-type: none"> <li>2. MPH risponde visualizzando l'interfaccia per la scelta del progetto nel quale caricare la release.</li> <li>3. Lo Studente sceglie il progetto e conferma.</li> <li>4. MPH risponde visualizzando l'interfaccia per l'upload della release.</li> <li>5. Lo Studente seleziona la release e conferma.</li> </ol>
Condizione d'uscita	6. MPH notifica allo Studente che l'upload è avvenuto con successo.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 6, MPH riscontra un ritardo nella consegna della release, notifica allo Studente la presenza di una penalità.</li> <li>• Nel caso in cui, al punto 6, MPH verifica problematiche legate all'upload della release, lo comunica allo Studente e il caso d'uso riprende dal punto 4.</li> </ul>

Tabella 4.13 – Caso d'uso: Upload release al progetto

Nome del caso d'uso	Visualizza release del progetto
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per accedere al progetto dove sono contenute le release di suo interesse.
Flusso degli eventi	<ol style="list-style-type: none"> <li>2. MPH risponde visualizzando l'interfaccia per la scelta del progetto del quale visualizzare le release.</li> <li>3. Lo Studente sceglie il progetto e conferma.</li> <li>4. MPH risponde visualizzando l'interfaccia con l'elenco delle release associate al progetto.</li> <li>5. Lo Studente seleziona la release.</li> </ol>
Condizione d'uscita	6. MPH mostra allo Studente la release selezionata.
Eccezioni	-----

Tabella 4.14 – Caso d'uso: Visualizza release del progetto

Nome del caso d'uso	Visualizza voto
Attori partecipanti	Studente
Condizione di ingresso	1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per accedere al progetto dove sono contenute le release valutate.



Flusso degli eventi	<ol style="list-style-type: none"> <li>2. Si procede secondo il caso d'uso Visualizza release del progetto.</li> <li>3. Lo Studente avvia la funzionalità di Visualizza voto per la release selezionata al punto 2.</li> </ol>
Condizione d'uscita	<ol style="list-style-type: none"> <li>4. MPH mostra allo Studente il voto richiesto.</li> </ol>
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH riscontra la mancanza del voto richiesto, lo comunica allo Studente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.15 – Caso d'uso: Visualizza voto

Nome del caso d'uso	Visualizza release condivise
Attori partecipanti	Studente
Condizione di ingresso	<ol style="list-style-type: none"> <li>1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per visualizzare le release condivise.</li> </ol>
Flusso degli eventi	<ol style="list-style-type: none"> <li>2. MPH risponde visualizzando l'interfaccia per visualizzare le release condivise.</li> <li>3. Lo Studente seleziona la release e conferma.</li> </ol>
Condizione d'uscita	<ol style="list-style-type: none"> <li>4. MPH mostra allo Studente la release selezionata.</li> </ol>
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH riscontra l'assenza di release condivise, lo comunica allo Studente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.16 – Caso d'uso: Visualizza release condivise

Nome del caso d'uso	Download release
Attori partecipanti	Studente
Condizione di ingresso	<ol style="list-style-type: none"> <li>1. Lo Studente avvia, dalla sua interfaccia, la funzionalità per effettuare il download di una data release condivisa.</li> </ol>
Flusso degli eventi	<ol style="list-style-type: none"> <li>2. Si procede secondo il caso d'uso Visualizza release condivise.</li> <li>3. Lo Studente avvia la funzionalità di Download per la release selezionata al punto 2.</li> </ol>
Condizione d'uscita	<ol style="list-style-type: none"> <li>4. MPH notifica allo Studente che il download si è concluso con successo.</li> </ol>
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH riscontra problematiche legate al download della release, lo comunica allo Studente e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.17 – Caso d'uso: Download release

### 4.2.3 Professore

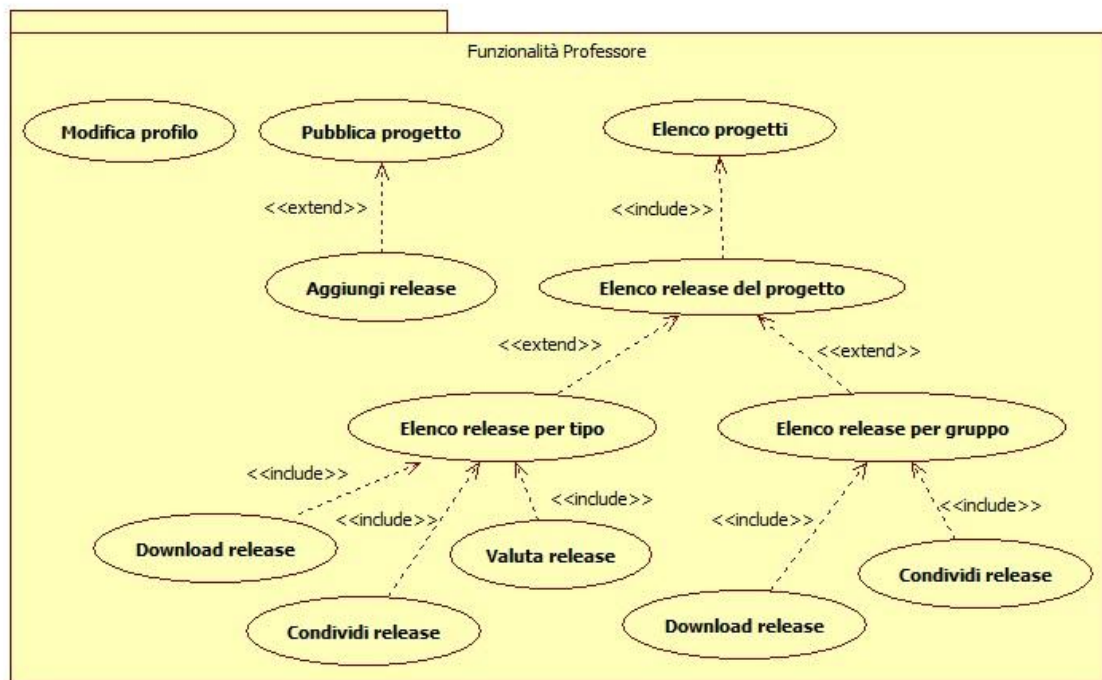


Figura 4.3 – Use Case Diagram: Professore

Nome del caso d'uso	Pubblica progetto
Attori partecipanti	Professore
Condizione di ingresso	1. Il Professore avvia, dalla sua interfaccia, la funzionalità per pubblicare un nuovo progetto.
Flusso degli eventi	2. MPH risponde visualizzando la form per l'inserimento del titolo, della descrizione e delle deadline. 3. Il Professore inserisce i dati e conferma.
Condizione d'uscita	4. MPH notifica al Professore che il progetto è stato pubblicato con successo.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 4, MPH verifica che il nome è già stato utilizzato, lo comunica al Professore e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.18 – Caso d'uso: Pubblica progetto

Nome del caso d'uso	Elenco release per tipo
Attori partecipanti	Professore

Condizione di ingresso	1. Il Professore avvia, dalla sua interfaccia, la funzionalità per accedere al progetto dove sono contenute le release di suo interesse.
Flusso degli eventi	2. MPH risponde visualizzando l'interfaccia per la scelta del progetto del quale ricercare la release. 3. Il Professore sceglie il progetto e conferma. 4. MPH risponde visualizzando l'interfaccia per la ricerca delle release con l'opzione di effettuare la ricerca in base al tipo di release o al gruppo. 5. Il Professore decide di effettuare la ricerca in base al tipo di release e conferma.
Condizione d'uscita	6. MPH mostra al Professore le release della categoria selezionata.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 6, MPH riscontra l'assenza di release della categoria selezionata, lo comunica al Professore e il caso d'uso riprende dal punto 4.</li> </ul>

Tabella 4.19 – Caso d'uso: Elenco release per tipo

Nome del caso d'uso	Elenco release per gruppo
Attori partecipanti	Professore
Condizione di ingresso	1. Il Professore avvia, dalla sua interfaccia, la funzionalità per accedere al progetto dove sono contenute le release di suo interesse.
Flusso degli eventi	2. MPH risponde visualizzando l'interfaccia per la scelta del progetto del quale ricercare la release. 3. Il Professore sceglie il progetto e conferma. 4. MPH risponde visualizzando l'interfaccia per la ricerca delle release con l'opzione di effettuare la ricerca in base al tipo di release o al gruppo. 5. Il Professore decide di effettuare la ricerca in base al gruppo e conferma.
Condizione d'uscita	6. MPH mostra al Professore le release del gruppo selezionato.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 6, MPH riscontra l'assenza di release del gruppo selezionato, lo comunica al Professore e il caso d'uso riprende dal punto 4.</li> </ul>

Tabella 4.20 – Caso d'uso: Elenco release per gruppo

Nome del caso d'uso	Condividi release
Attori partecipanti	Professore
Condizione di ingresso	1. Il Professore avvia dalla sua interfaccia, la funzionalità per accedere al progetto nel quale sono contenute le release da condividere.
Flusso degli eventi	2. Si procede secondo il caso d'uso Elenco release per tipo o Elenco release per gruppo. 3. Il Professore seleziona la release, abilita la condivisione della stessa a uno o più gruppi e conferma.
Condizione d'uscita	4. MPH notifica al Professore che la condivisione si è conclusa con successo.
Eccezioni	-----

Tabella 4.21 – Caso d'uso: Condividi release

Nome del caso d'uso	Valuta release
Attori partecipanti	Professore
Condizione di ingresso	1. Il Professore avvia, dalla sua interfaccia, la funzionalità per accedere al progetto nel quale sono contenute le release da valutare.
Flusso degli eventi	2. Si procede secondo il caso d'uso Elenco release per tipo. 3. Il Professore seleziona la release, assegna un voto e conferma.
Condizione d'uscita	4. MPH notifica al Professore che la registrazione del voto si è conclusa con successo.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH riscontra problematiche legate alla registrazione del voto, lo comunica al Professore e il caso d'uso riprende dal punto 3.</li> </ul>

Tabella 4.22 – Caso d'uso: Valuta release

Nome del caso d'uso	Download release
Attori partecipanti	Professore
Condizione di ingresso	1. Il Professore avvia, dalla sua interfaccia, la funzionalità per accedere al progetto nel quale sono contenute le release da scaricare.
Flusso degli eventi	2. Si procede secondo il caso d'uso Elenco release per tipo o Elenco release per gruppo.

	3. Il Professore seleziona la release, sceglie di effettuarne il download e conferma.
Condizione d'uscita	4. MPH notifica al Professore che il download della release si è concluso con successo.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 4, MPH riscontra problematiche legate al download della release selezionata, lo comunica al Professore e il caso d'uso riprende dal punto 3.</li> </ul>

Tabella 4.23 – Caso d'uso: Download release

## 4.2.4 Amministratore

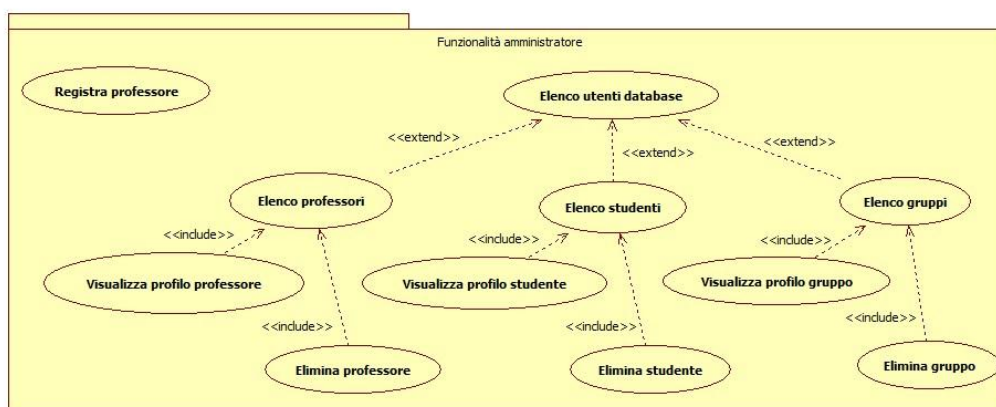


Figura 4.4 – Use Case Diagram: Amministratore

Nome del caso d'uso	Registra professore
Attori partecipanti	Amministratore
Condizione di ingresso	1. L'Amministratore avvia, dalla sua interfaccia, la funzionalità per registrare un nuovo professore.
Flusso degli eventi	2. MPH risponde visualizzando la form di inserimento dei dati generali del professore. 3. L'Amministratore inserisce i dati e conferma.
Condizione d'uscita	4. MPH verifica che lo username scelto dall'Amministratore non sia già in uso e conferma l'avvenuta registrazione.
Eccezioni	<ul style="list-style-type: none"> <li>Nel caso in cui, al punto 4, MPH verifica che lo username scelto è già in uso, lo comunica all'Amministratore e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.24 – Caso d'uso: Registra professore

Nome del caso d'uso	Visualizza profilo professore
Attori partecipanti	Amministratore
Condizione di ingresso	1. L'Amministratore avvia, dalla sua interfaccia, la funzionalità per accedere all'elenco degli utenti presenti nel sistema.
Flusso degli eventi	2. MPH risponde visualizzando l'elenco degli utenti presenti nel sistema. 3. L'Amministratore decide di visualizzare il profilo di un professore.
Condizione d'uscita	4. MPH risponde visualizzando il profilo richiesto dall'Amministratore.
Eccezioni	-----

Tabella 4.25 – Caso d'uso: Visualizza profilo professore

Nome del caso d'uso	Elimina professore
Attori partecipanti	Amministratore
Condizione di ingresso	1. L'Amministratore avvia, dalla sua interfaccia, la funzionalità per accedere all'elenco degli utenti presenti nel sistema.
Flusso degli eventi	2. MPH risponde visualizzando l'elenco degli utenti presenti nel sistema. 3. L'Amministratore decide di eliminare un professore dal portale.
Condizione d'uscita	4. MPH notifica all'Amministratore che l'eliminazione del professore si è conclusa con successo.
Eccezioni	<ul style="list-style-type: none"> <li>• Nel caso in cui, al punto 4, MPH riscontra problematiche legate all'eliminazione del profilo del professore, lo comunica all'Amministratore e il caso d'uso riprende dal punto 2.</li> </ul>

Tabella 4.26 – Caso d'uso: Elimina professore

### 4.3 Diagrammi

I modelli dei casi d'uso risultano costituiti da due proiezioni: la proiezione statica e la proiezione dinamica. La prima viene completamente catturata attraverso gli Use Case Diagram (esposti e rappresentati nel paragrafo 4.2), mentre per la seconda sono disponibili opportuni diagrammi UML, quali Sequence Diagram e Activity

Diagram. Per poter correttamente passare dalla visione statica a quella dinamica è importante realizzare il Class Diagram, in quanto mette in evidenza gli elementi base del sistema racchiusi con il nome di classe.

### 4.3.1 Class Diagram

I diagrammi delle classi (Class Diagram) consentono di descrivere le entità, mettendo in luce i loro attributi e le eventuali relazioni che si generano tra di esse.

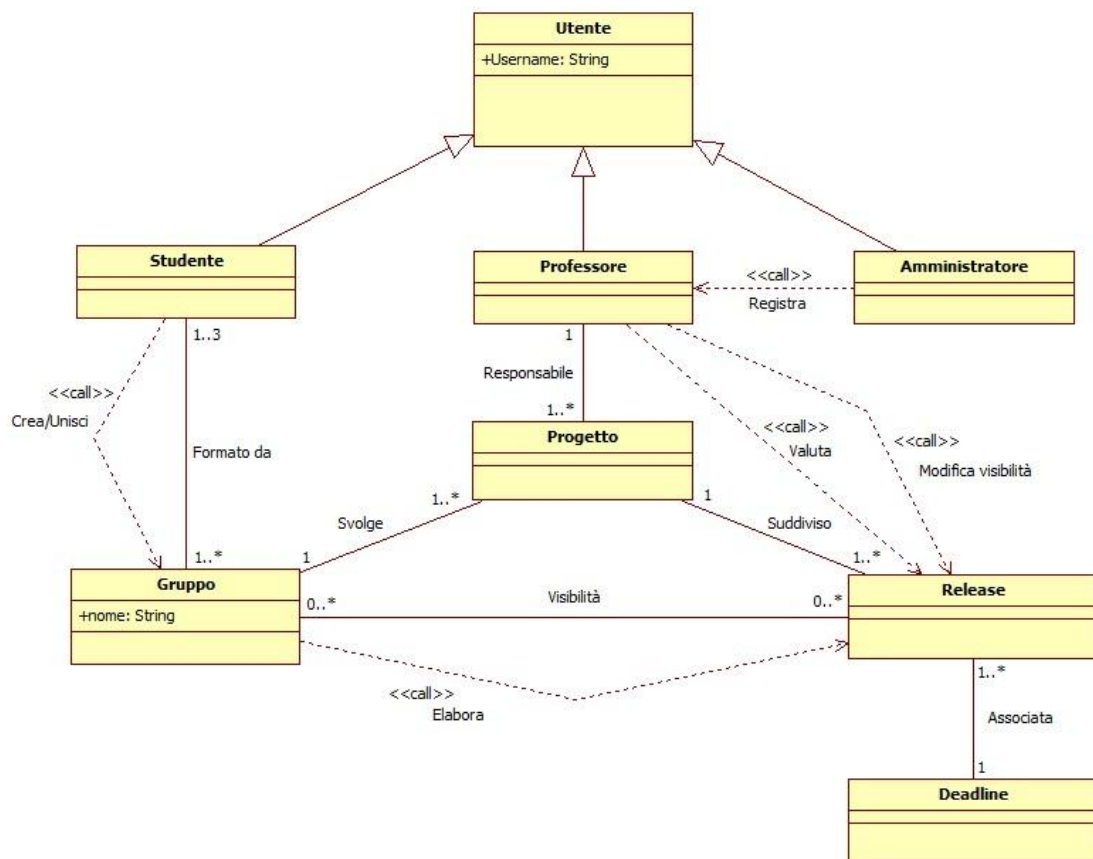


Figura 4.5 – Class Diagram

### 4.3.2 Sequence Diagram

Il Diagramma di Sequenza (Sequence Diagram) è un diagramma previsto dall'UML atto a descrivere uno specifico scenario; il suo compito è quello di enfatizzare le azioni che si svolgono tra le diverse entità.

- Registrazione studente

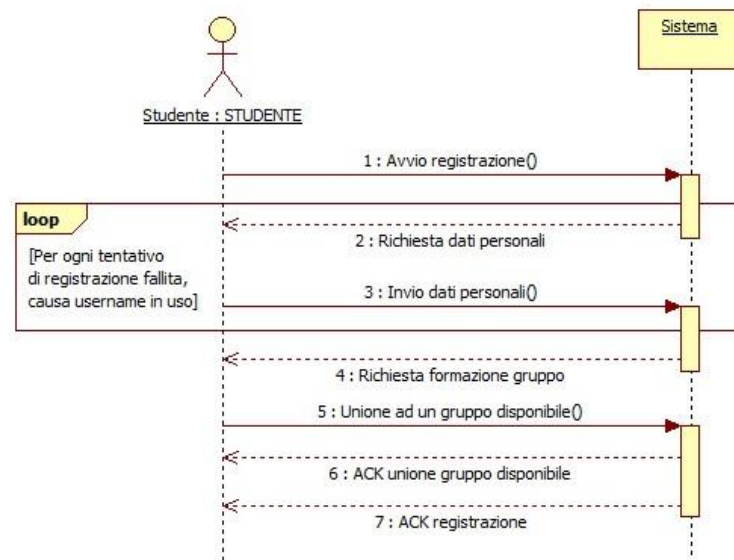


Figura 4.6 – Sequence Diagram: Registrazione studente

- Creazione nuovo gruppo

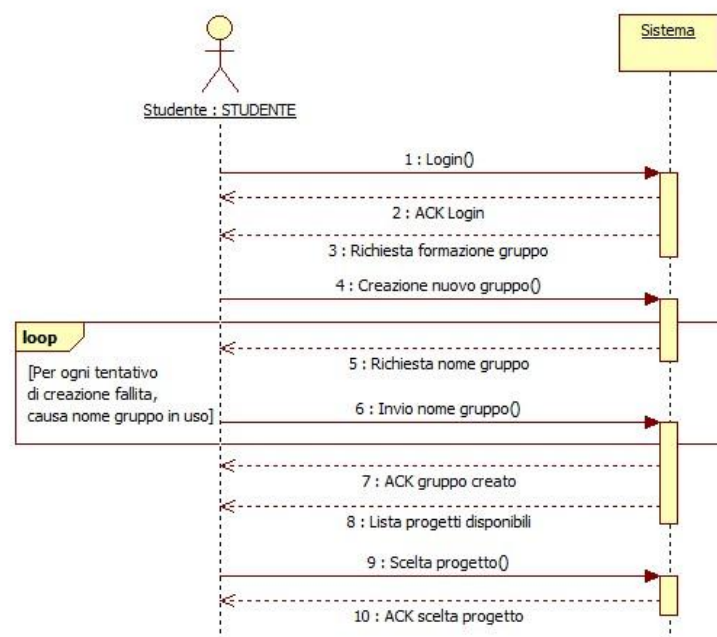


Figura 4.7 – Sequence Diagram: Creazione nuovo gruppo



- Upload release

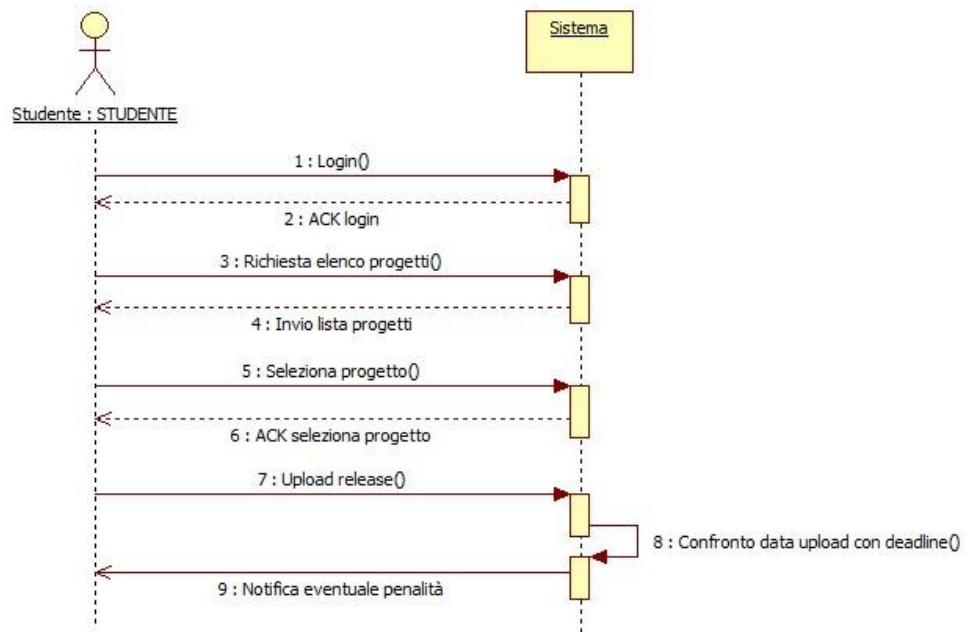


Figura 4.8 – Sequence Diagram: Upload release

- Valutazione release

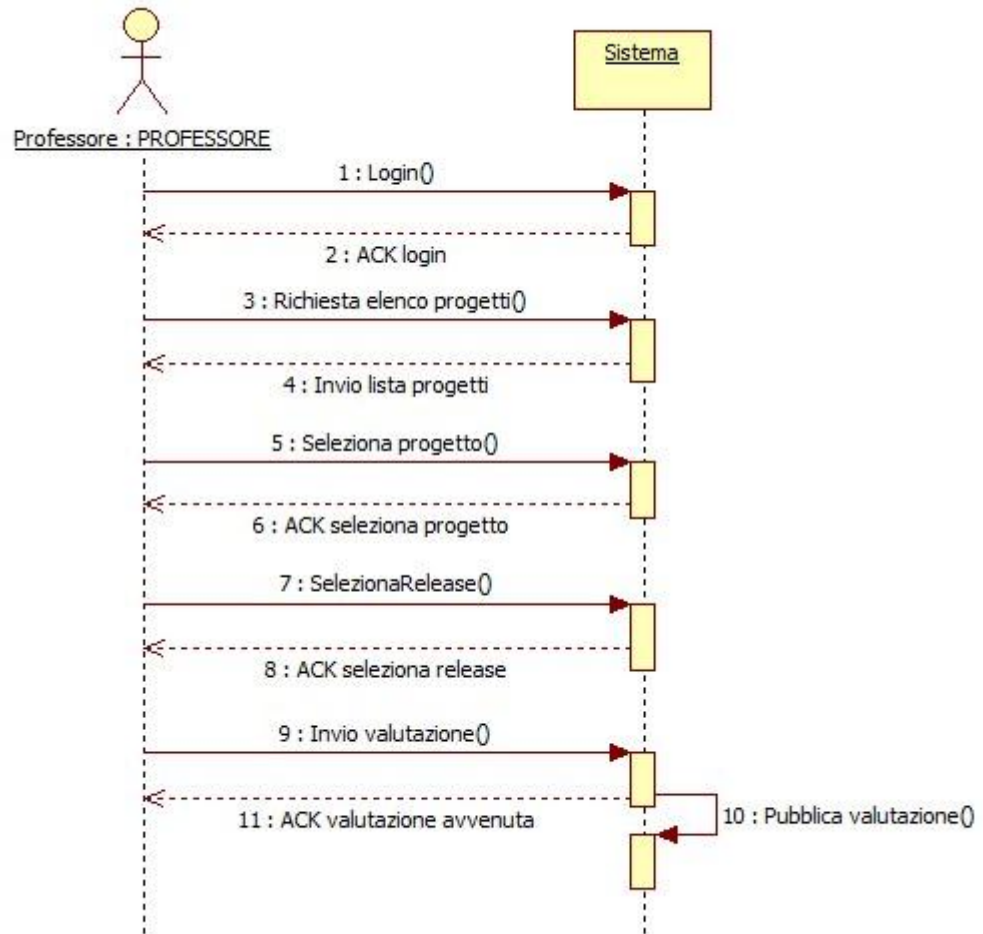


Figura 4.9 – Sequence Diagram: Valutazione release

- Condividi release

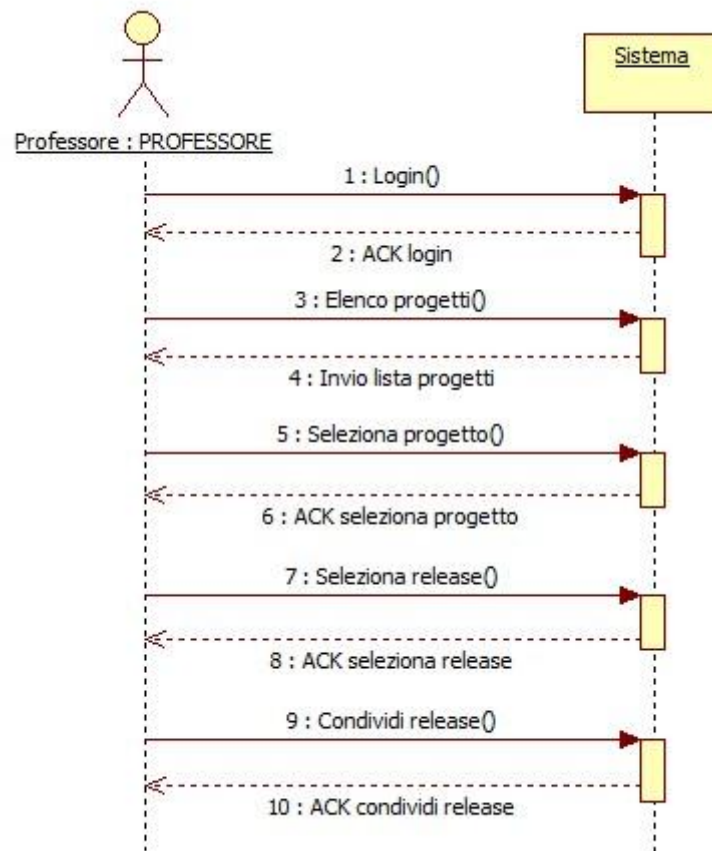


Figura 4.10 – Sequence Diagram: Condividi release

### 4.3.3 Activity Diagram

Il Diagramma delle Attività (Activity Diagram) è un diagramma utilizzato per descrivere una sequenza di operazioni il cui svolgimento permette la definizione di un'attività più complessa. È spesso usato come modello complementare del Sequence Diagram per descrivere il comportamento dinamico del sistema.

- Registrazione studente

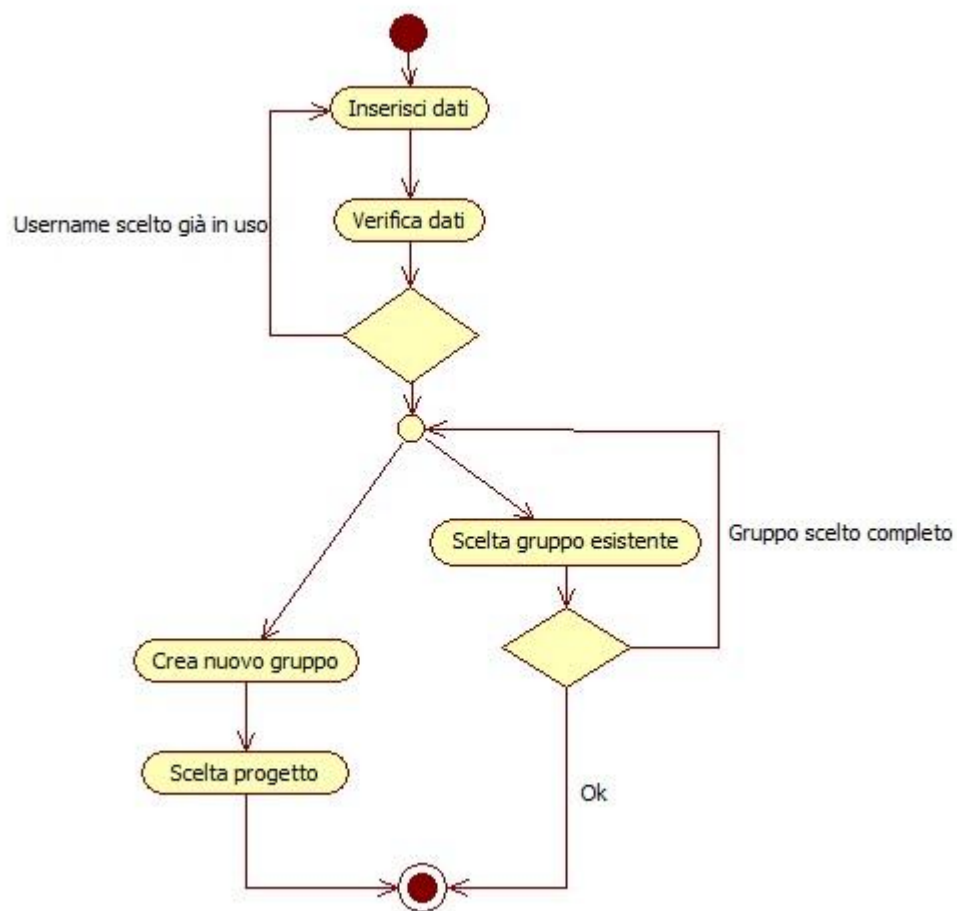


Figura 4.11 – Activity Diagram: Registrazione studente

- Gestione release

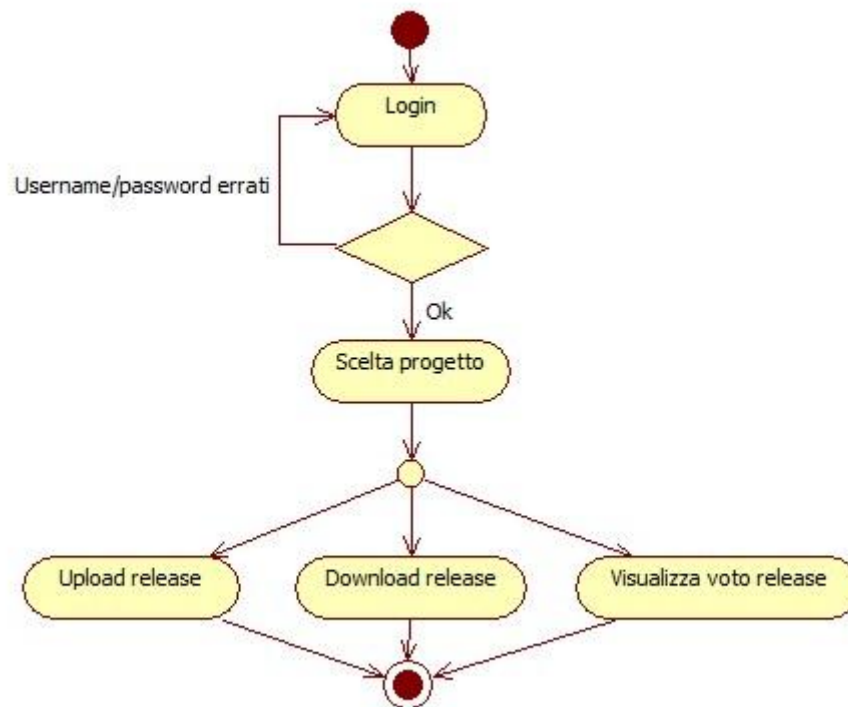


Figura 4.12 – Activity Diagram: Gestione release

- Valutazione release



Figura 4.13 – Activity Diagram: Valutazione release

- Pubblica progetto

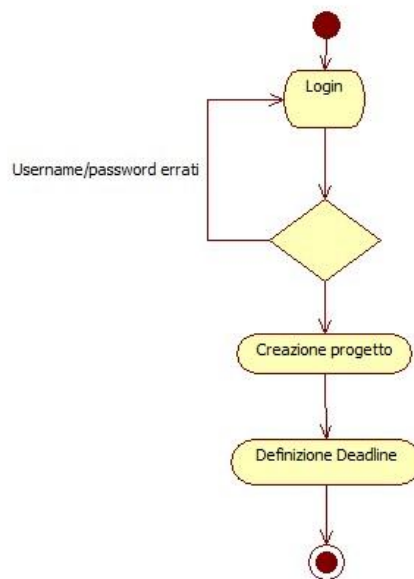


Figura 4.14 – Activity Diagram: Pubblica progetto



## SPECIFICHE

**I**n questo capitolo verranno riportate le specifiche che garantiscono il corretto funzionamento del sistema sulla base dei requisiti raccolti e presentati nel capitolo 3.

### 5.1 Definizione delle specifiche

Le specifiche del sistema possono riassumersi nei seguenti punti:

- Ogni Utente deve avere uno Username univoco.
- Ogni Studente deve avere una matricola univoca.
- Ogni Studente si registra autonomamente al sistema.
- Ogni Studente, dopo essersi registrato, sceglie il Gruppo di lavoro.
- Il Professore è registrato al sistema dall'Amministratore.
- Ogni Studente appartiene ad uno o più Gruppi.
- Ad ogni Gruppo appartengono 1, 2 o 3 Studenti.
- Ogni Gruppo deve avere un nome univoco.
- Ogni Gruppo svolge uno o più Progetti.
- Ogni Release è associata ad un solo Gruppo.
- Ogni Release può essere visibile a uno o più Gruppi.

- Ogni Release deve essere consegnata entro la Deadline corrispondente.
- Se la Release viene consegnata successivamente alla Deadline il sistema attribuisce al gruppo corrispondente una penalità.
- Ad ogni Release deve essere associato un solo Voto parziale.
- Il Voto parziale è un numero da 1 a 10.
- Il Voto parziale è costituito dalla differenza tra la valutazione proposta dal Professore e l'eventuale penalità attribuita dal sistema.
- Le Release non consegnate hanno Voto parziale pari a 1.
- Ad ogni Gruppo deve essere associato un solo Voto finale, a fine progetto.
- Il Voto finale, calcolato dal sistema, è dato dalla media dei Voti parziali.

## 5.2 Verifica delle specifiche

Per verificare il modello di sistema definito dalle specifiche (esposte al paragrafo 5.1) si è utilizzato un modello Alloy. Di seguito verrà riportato interamente il codice del modello, suddiviso in Signatures, Facts, Assertion & Predicates e il mondo generato dallo strumento.

Nel mondo creato sono stati riportati solamente gli attributi Username per Utente, Nome per Gruppo e Nome per Progetto, in quanto rappresentano gli elementi identificativi delle rispettive entità su cui sono stati opportunamente determinati i vincoli essenziali.

### 5.2.1 Signatures

Le signatures rappresentano le entità significative del modello del sistema.

```
module MPH

sig Stringa{}

abstract sig Utente{
    username: Stringa
}
```



```

one sig Amministratore extends Utente{}

sig Studente extends Utente{
    appartiene: some Gruppo
}

sig Professore extends Utente{
    responsabile: some Progetto
}

sig Gruppo{
    nome: Stringa,
    haVisibilita: set Release,
    formatoDa: some Studente,
    svolge: some Progetto
} {#formatoDa <= 3}

sig Progetto{
    nome: Stringa,
    seguito: one Professore,
    vieneSvolto: one Gruppo,
    suddiviso: some Release
}

sig Release {
    visibile: some Gruppo,
    parteDi: one Progetto
}

```

## 5.2.2 Facts

I facts sono utili per descrivere in modo esaustivo le proprietà del sistema.

```

// Ogni Utente deve avere uno Username univoco.
fact UsernameUnico{
    all u1, u2: Utente |
        (u1.username = u2.username) <=> (u1 = u2)
}

// Ogni Gruppo deve avere un nome univoco.
fact NomeUnicoGruppo{
    all g1, g2: Gruppo |
        (g1.nome = g2.nome) <=> (g1 = g2)
}

```

```

// Ogni Progetto deve avere un nome univoco.
fact NomeUnicoProgetto{
    all p1, p2: Progetto |
        (p1.nome = p2.nome) <=> (p1 = p2)
}

// Ogni Progetto deve far riferimento ad un solo Professore.
fact OgniProgettoUnProfessore{
    all pg: Progetto | one pf: Professore | pg in pf.responsabile
}

// Se uno Studente appartiene ad un Gruppo, allora quel Gruppo ha
// tra i suoi membri lo Studente stesso.
fact StudenteGruppo{
    all g: Gruppo,
    s: Studente |
        (s in g.formatoDa) <=> (g in s.appartiene)
}

// Se un Progetto è tra quelli seguiti da un Professore, allora il
// Professore è responsabile di quel Progetto.
fact ProfessoreProgetto{
    all pf: Professore,
    pg: Progetto |
        (pg in pf.responsabile) <=> (pf = pg.seguito)
}

// Un Progetto non può essere associato a due Professori diversi.
fact NoProgettoCon2Professori{
    all pf1, pf2: Professore,
    pg: Progetto |
        (pg in pf1.responsabile && pg in pf2.responsabile) =>
            (pf1 = pf2)
}

// Due Gruppi non possono svolgere lo stesso Progetto.
fact NoGruppiStessoProgetto{
    all g1, g2: Gruppo |
        (g1.svolge = g2.svolge) <=> (g1 = g2)
}

// Se un Progetto è svolto da un Gruppo, allora quel Gruppo ha tra
// la lista dei suoi lavori il Progetto stesso.
fact ProgettoGruppo{
    all p: Progetto,
    g: Gruppo |
        (p.vieneSvolto = g) <=> (p in g.svolge)
}

```

```

// Un Progetto non può essere svolto da due Gruppi diversi.
fact NoProgettoIn2Gruppi{
    all p1, p2: Progetto |
        (p1.vieneSvolto = p2.vieneSvolto) <=> (p1 = p2)
}

// Se una Release è parte di un Progetto, allora quel Progetto
// presenta tra le sue consegne la Release stessa.
fact ReleaseProgetto{
    all p: Progetto,
        r: Release |
        (r in p.suddiviso) <=> (p = r.parteDi)
}

// Se la Release è abilitata per un Gruppo, allora quel Gruppo può
// vedere la Release stessa.
fact ReleaseGruppo{
    all g: Gruppo,
        r: Release |
        (r in g.haVisibilita) <=> (g in r.visibile)
}

// Le Release abilitate alla condivisione non sono le Release
// presentate dal Gruppo stesso.
fact ReleaseVisibiliNonDelGruppo{
    all g: Gruppo,
        r1, r2: Release |
        (r1 in g.haVisibilita && r2 in g.svolge.suddiviso) =>
        (r1 != r2)
}

```

### 5.2.3 Assertions & Predicates

Le assertions e i predicates servono a verificare la bontà del modello. In particolare si vuole enfatizzare come gli scenari, descritti dettagliatamente nel capitolo 4, siano compatibili con il modello generato.

```

// Verifico che non ci siano due Utenti con lo stesso Username.
assert ContolloUsername{
    (no u1, u2: Utente | u1 != u2 && u1.username = u2.username)
}
check ContolloUsername for 10

```

```

// Verifico che non ci siano due Gruppi con lo stesso nome.
assert ContolloNomeGruppo{
    (no g1, g2: Gruppo | g1 != g2 && g1.nome = g2.nome)
}
check ContolloNomeGruppo for 10

// Verifico che non ci siano due Progetti con lo stesso nome.
assert ContolloNomeProgetto{
    (no p1, p2: Progetto | p1 != p2 && p1.nome = p2.nome)
}
check ContolloNomeProgetto for 10

// Verifico che non ci sia uno Studente che non appartiene a
// nessun Gruppo.
assert ContolloNoStudenteSenzaGruppo{
    (no s: Studente | #s.appartiene = 0)
}
check ContolloNoStudenteSenzaGruppo for 10

//Verifico che un Gruppo sia formato da 1, 2 o 3 Studenti.
assert ContolloGruppo{
    (no g: Gruppo | #g.formatoDa > 3 && #g.formatoDa = 0)
}
check ContolloGruppo for 10

// Verifico che non ci sia un Gruppo non stia svolgendo nessun
// Progetto.
assert ContolloNoGruppoSenzaProgetto{
    (no g: Gruppo | #g.svolge = 0)
}
check ContolloNoGruppoSenzaProgetto for 10

// Verifico che non ci sia un Progetto che non è seguito da nessun
// Professore.
assert ContolloNoProgettoSenzaProfessore{
    (no pg: Progetto | #pg.seguito = 0)
}
check ContolloNoProgettoSenzaProfessore for 10

// Verifico che non ci sia una Release svolta da un Gruppo non
// rientra tra le Release di cui quel Gruppo ha visibilità.
assert ContolloNoVisibilitaReleasePerLeReleaseDelGruppo{
    (no r: Release, g:Gruppo | r in g.haVisibilita &&
    r in g.svolge.suddiviso)
}
check ContolloNoVisibilitaReleasePerLeReleaseDelGruppo for 10

```

```

// Verifico che non ci sia un Progetto che viene seguito da due
// Professori diversi.
assert ControlloProgettoCon2Professori{
    (no pf1, pf2: Professore, pg: Progetto | pf1 != pf2 &&
     pg in pf1.responsabile && pg in pf2.responsabile)
}
check ControlloProgettoCon2Professori for 10

// Verifico che non ci sia un Progetto che viene svolto da due
// Gruppi diversi.
assert ControlloProgettoIn2Gruppi{
    (no g1, g2: Gruppo, p: Progetto | g1 != g2 &&
     p in g1.svolge && p in g2.svolge)
}
check ControlloProgettoIn2Gruppi for 10

// Predicato per visualizzare il mondo
pred showMPH{
    #Release = 3
    #Professore = 2
    #Gruppo = 2
    #Studente = 4
    #Progetto = 3
    some Stringa
}
run showMPH for 10

```

## 5.2.4 Mondo generato


Per la verifica del modello generato, presentato alla pagina seguente, si procede con la dimostrazione scritte delle Assertions più significative.

Controllo Username univoco per Utente (  : username):


- Professore0 ha Username in Stringa2
- Professore1 ha Username in Stringa4
- Studente0 ha Username in Stringa1
- Studente1 ha Username in Stringa8
- Studente2 ha Username in Stringa9
- Studente3 ha Username in Stringa3
- Amministratore ha Username in Stringa0

Controllo Nome univoco per Gruppo (  : nome):

- Gruppo0 ha Nome in Stringa9
- Gruppo1 ha Nome in Stringa8

Controllo Nome univoco per Progetto (  : nome):


- Progetto0 ha Nome in Stringa7
- Progetto1 ha Nome in Stringa6

Controllo cardinalità del Gruppo (  : formatoDa):




- Gruppo0 formato da Studente2, Studente3
- Gruppo1 formato da Studente0, Studente1, Studente3

Controllo Progetto seguito da un solo Professore (  : seguito):

- Progetto0 seguito da Professore1
- Progetto1 seguito da Professore0

Controllo stessa Release non si riferisca a due Progetti diversi (  : parteDi):

- Release0 è parte di Progetto1
- Release1 è parte di Progetto1
- Release2 è parte di Progetto0

Controllo Release visibile dal Gruppo solo se Release non appartiene al Progetto seguito dal Gruppo (  : haVisibilita,  : svolge,  : suddiviso):

- Gruppo0 ha visibilità su Release2 (perché svolge Progetto1, il quale è suddiviso in Release0 e Release1)
- Gruppo1 ha visibilità su Release0 e Release 1 (perché svolge Progetto0, il quale è suddiviso in Release2)

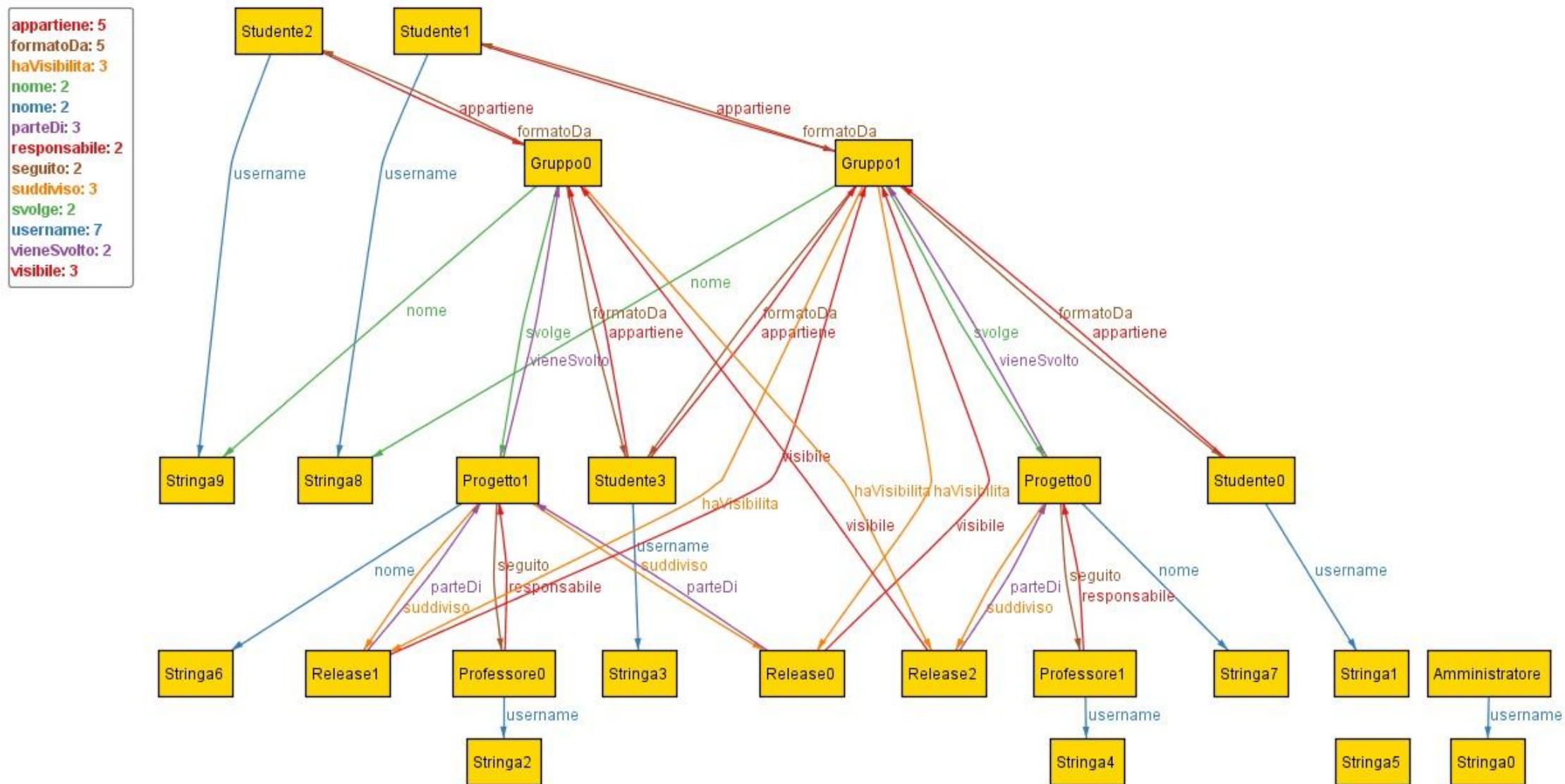


Figura 5.1 – Mondo generato Alloy



## SOFTWARE UTILIZZATI

**I**n questa sezione verranno brevemente definiti i software utilizzati nella realizzazione di questo documento.

### A.1 StarUML

StarUML è una piattaforma open source utilizzata per lo sviluppo veloce di diagrammi UML, secondo il paradigma MDA. È flessibile ed estensibile grazie alla sua architettura a plug-in e alla disponibilità delle apposite API.

All'interno di questo documento starUML è stato utilizzato per la composizione dello Use Case Diagram, Class Diagram, Sequence Diagram e Activity Diagram.

### A.2 Alloy Analyzer 4.0

Alloy Analyzer è uno strumento utilizzato per la verifica automatica dei modelli Alloy (file.als). A seconda della struttura definita, sarà possibile sfruttare questo tool per cercare istanze oppure sfidarlo a trovare un controesempio che smentisce determinate asserzioni, nei limiti di uno scope (ossia il massimo numero di atomi di tipo top-level che si vogliono utilizzare).