

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



## Progetto di Ingegneria del Software 2

### Parte II: DD 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
1. Database .....	5
1.1 Modello E/R .....	5
1.1.1 Entità .....	5
1.1.2 Associazioni .....	7
1.2 Modello relazionale .....	8
2. Modelli .....	11
2.1 Modelli di navigazione .....	11
2.1.1 Studente .....	12
2.1.2 Professore .....	13
2.1.3 Amministratore .....	14
2.2 Diagrammi di analisi .....	15
2.2.1 Studente .....	15
2.2.2 Professore .....	16
2.2.3 Amministratore .....	17
2.3 Diagrammi di dettaglio .....	18
2.3.1 Login .....	18
2.3.2 Studente .....	19
2.3.3 Professore .....	20
2.3.4 Amministratore .....	21

Appendice A. Software utilizzati .....	22
A.1 StarUML.....	22
A.2 MySQL.....	22
A.3 Microsoft Visio 2010 .....	22

## ELENCO DELLE FIGURE

Fig. 1.1 – Modello E/R.....	7
Fig. 1.2 – Modello Relazionale .....	10
Fig. 2.1 – UX Model: Studente.....	12
Fig. 2.2 – UX Model: Professore .....	13
Fig. 2.3 – UX Model: Amministratore .....	14
Fig. 2.4 – BCE Model: Studente .....	15
Fig. 2.5 – BCE Model: Professore .....	16
Fig. 2.6 – BCE Model: Amministratore .....	17
Fig. 2.7 – JSP Model: Login .....	18
Fig. 2.8 – JSP Model: Studente.....	19
Fig. 2.9 – JSP Model: Professore .....	20
Fig. 2.10 – JSP Model: Amministratore .....	21



# DATABASE

**P**er la gestione dei dati persistenti è stata realizzata una base di dati relazionale, alla quale si accede con l'apposito sistema di identificazione. A tal proposito verranno di seguito mostrati il modello E/R e lo schema relazionale del database.

## 1.1 Modello E/R

Il modello E/R è utilizzato per la rappresentazione concettuale dei dati ad un alto livello di astrazione. Per la sua semplicità intuitiva è spesso utilizzato nelle prime fasi della progettazione di una base di dati per tradurre i risultati derivanti dall'analisi di un determinato dominio in uno schema concettuale.

### 1.1.1 Entità

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

- **Utente**

È l'entità padre delle entità *Studente*, *Professore* e *Amministratore*. I suoi attributi sono: *Username*, *Password*, *Nome*, *Cognome*, *Email*. L'identificatore interno è *Username*.

- **Studente**  
È l'entità figlia dell'entità Utente. I suoi attributi sono quelli ereditati dall'entità padre ai quali si aggiunge: *Matricola*.
- **Professore**  
È l'entità figlia dell'entità Utente. I suoi attributi sono quelli ereditati dall'entità padre ai quali si aggiunge: *Telefono*.
- **Amministratore**  
È l'entità figlia dell'entità Utente. I suoi attributi sono quelli ereditati dall'entità padre.
- **Gruppo**  
Possiede gli attributi: *Nome, Control*. L'identificatore interno è *Nome*.
- **Progetto**  
Possiede gli attributi: *Nome, Descrizione, Materia*. L'identificatore interno è *Nome*.
- **Release**  
Possiede gli attributi: *Id, Tipo, Deadline, Consegnabile*. L'identificatore interno è *Id*.
- **File**  
Possiede gli attributi: *Id, Url, Descrizione*. L'identificatore interno è *Id*.
- **Progetto\_Gruppo**  
È l'entità debole che associa un progetto con il gruppo che lo sta svolgendo.  
Possiede gli attributi: *Id, VotoFinale*. L'identificatore interno è *Id*.
- **Progetto\_Release**  
È l'entità debole che mostra le release consegnate da un gruppo, il quale sta svolgendo un determinato progetto. Possiede gli attributi: *Id, VotoParziale, Penalita, DataUpload*. L'identificatore interno è *Id*.

## 1.1.2 Associazioni

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

- **Formato da** lega le entità Studente e Gruppo.
- **Responsabile** lega le entità Professore e Progetto.
- **Svolge** lega le entità Gruppo e Progetto\_Gruppo.
- **Suddiviso** lega le entità Progetto\_Release e Progetto\_Gruppo.
- **Appartiene** lega le entità Progetto e Progetto\_Gruppo.
- **Composto** lega le entità Progetto e Release.
- **Tipo** lega le entità Release e Progetto\_Release.
- **Fa parte** lega le entità Progetto\_Release e File.
- **Ha visibilità** lega le entità Progetto\_Release e Gruppo.

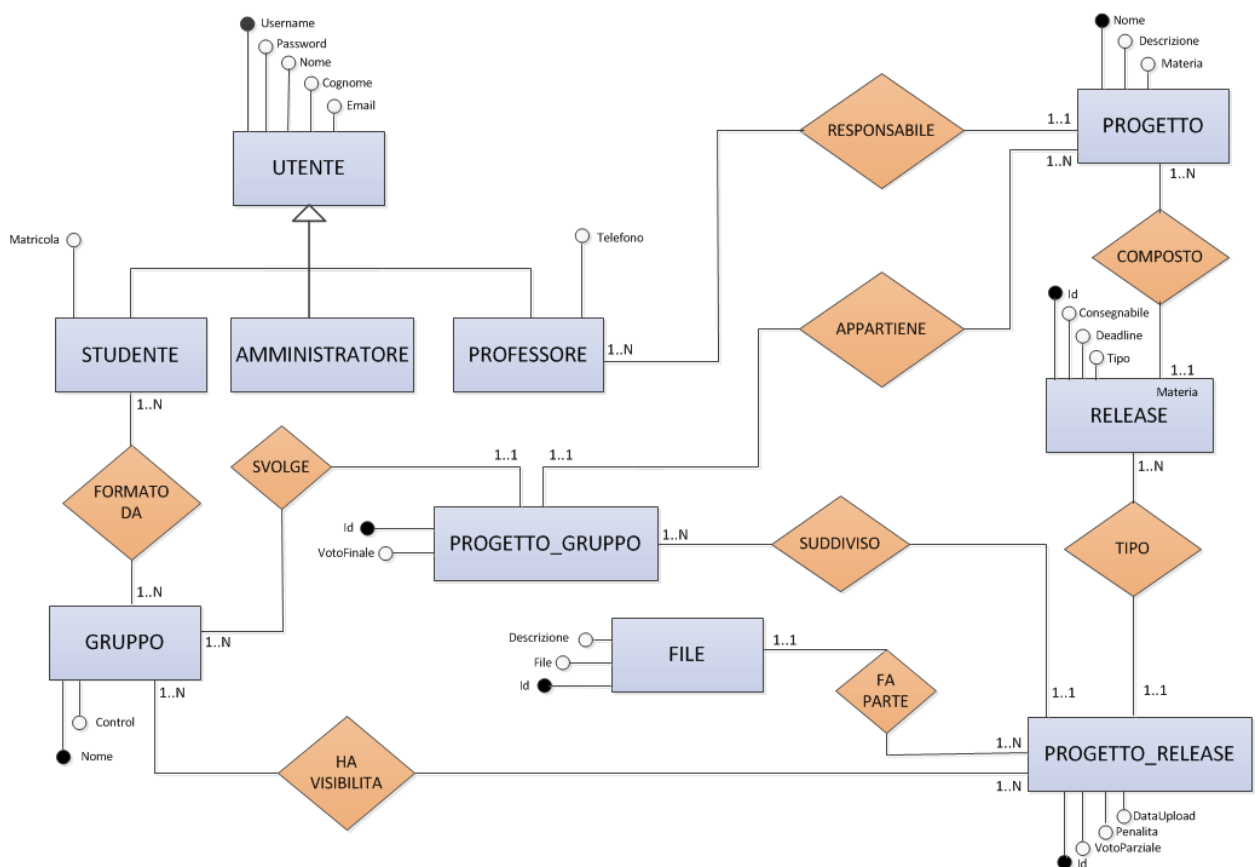


Figura 1.1 – Modello E/R

## 1.2 Modello Relazionale

Il modello relazionale è utilizzato per la rappresentazione logica dei dati con l'obiettivo di costituire il database per l'effettiva realizzazione dell'applicazione, anche a costo di una ristrutturazione forzata del modello concettuale.

L'associazione uno a molti tra le entità **Professore** e **Progetto** suggerisce l'eliminazione della relazione **Responsabile**, con l'accorpamento dell'attributo *UsernameProfessore* all'entità **Progetto**.

L'associazione uno a molti tra le entità **Progetto** e **Release** suggerisce l'eliminazione della relazione **Composto**, con l'accorpamento dell'attributo *NomeProgetto* all'entità **Release**.

L'associazione uno a molti tra le entità **Gruppo** e **Progetto\_Gruppo** suggerisce l'eliminazione della relazione **Svolge**, con l'accorpamento dell'attributo *NomeGruppo* all'entità **Progetto\_Gruppo**.

L'associazione uno a molti tra le entità **Progetto** e **Progetto\_Gruppo** suggerisce l'eliminazione della relazione **Appartiene**, con l'accorpamento dell'attributo *NomeProgetto* all'entità **Progetto\_Gruppo**.

L'associazione uno a molti tra le entità **Progetto\_Release** e **File** suggerisce l'eliminazione della relazione **Fa parte**, con l'accorpamento dell'attributo *IdProgettoRelease* all'entità **File**.

L'associazione uno a molti tra le entità **Progetto\_Release** e **Release** suggerisce l'eliminazione della relazione **Tipo**, con l'accorpamento dell'attributo *IdRelease* all'entità **Progetto\_Release**.

L'associazione uno a molti tra le entità **Progetto\_Release** e **Progetto\_Gruppo** suggerisce l'eliminazione della relazione **Suddiviso**, con l'accorpamento dell'attributo *IdProgettoGruppo* all'entità **Progetto\_Release**.

L'associazione molti a molti tra le entità **Studente** e **Gruppo** suggerisce la definizione della tabella **Studente\_Gruppo** avente come attributi gli identificatori interni *UsernameStudente*, *NomeGruppo* delle entità **Studente** e **Gruppo** in qualità di identificatori esterni.



L'associazione molti a molti tra le entità **Gruppo** e **Progetto\_Release** suggerisce la definizione della tabella **Release\_Condivisa** avente come attributi gli identificatori interni IdProgettoRelease, NomeGruppoProp delle entità **Gruppo** e **Progetto\_Release** in qualità di identificatori esterni.

Lo schema relazionale si può sintetizzare come segue:

- **Studiante**(Username, Password, Nome, Cognome, Email, Matricola)
- **Professore**(Username, Password, Nome, Cognome, Email, Telefono)
- **Amministratore**(Username, Password, Nome, Cognome, Email)
- **Gruppo**(Nome, Control)
- **Studiante\_Gruppo**(UsernameStudiante, NomeGruppo)
- **Progetto**(Nome, Descrizione, Materia, UsernameProfessore)
- **Progetto\_Gruppo**(Id, VotoFinale, NomeProgetto, NomeGruppo)
- **Progetto\_Release**(Id, VotoParziale, Penalita, DataUpload, IdProgettoGruppo, IdRelease)
- **Release**(Id, Tipo, Deadline, Consegnabile, NomeProgetto)
- **Release\_Condivisa**(IdProgettoRelease, NomeGruppoProp)
- **File**(Id, Url, Descrizione, IdProgettoRelease)





## MODELLI

**I**n questo capitolo vengono descritti i modelli su cui si basa l'analisi del sistema. In particolare verranno mostrati i modelli di navigazione, i diagrammi di analisi e i diagrammi di dettaglio. Per non appesantire la lettura, in alcuni schemi sono stati omessi i metodi `get()` e `set()` legati agli attributi delle entity.

### 2.1 Modelli di navigazione

Per illustrare i percorsi di navigazione dell'applicazione sono stati realizzati i seguenti modelli di navigazione, chiamati UX Diagram (User eXperience Diagram), organizzati in relazione al tipo di utente.

### 2.1.1 Studente

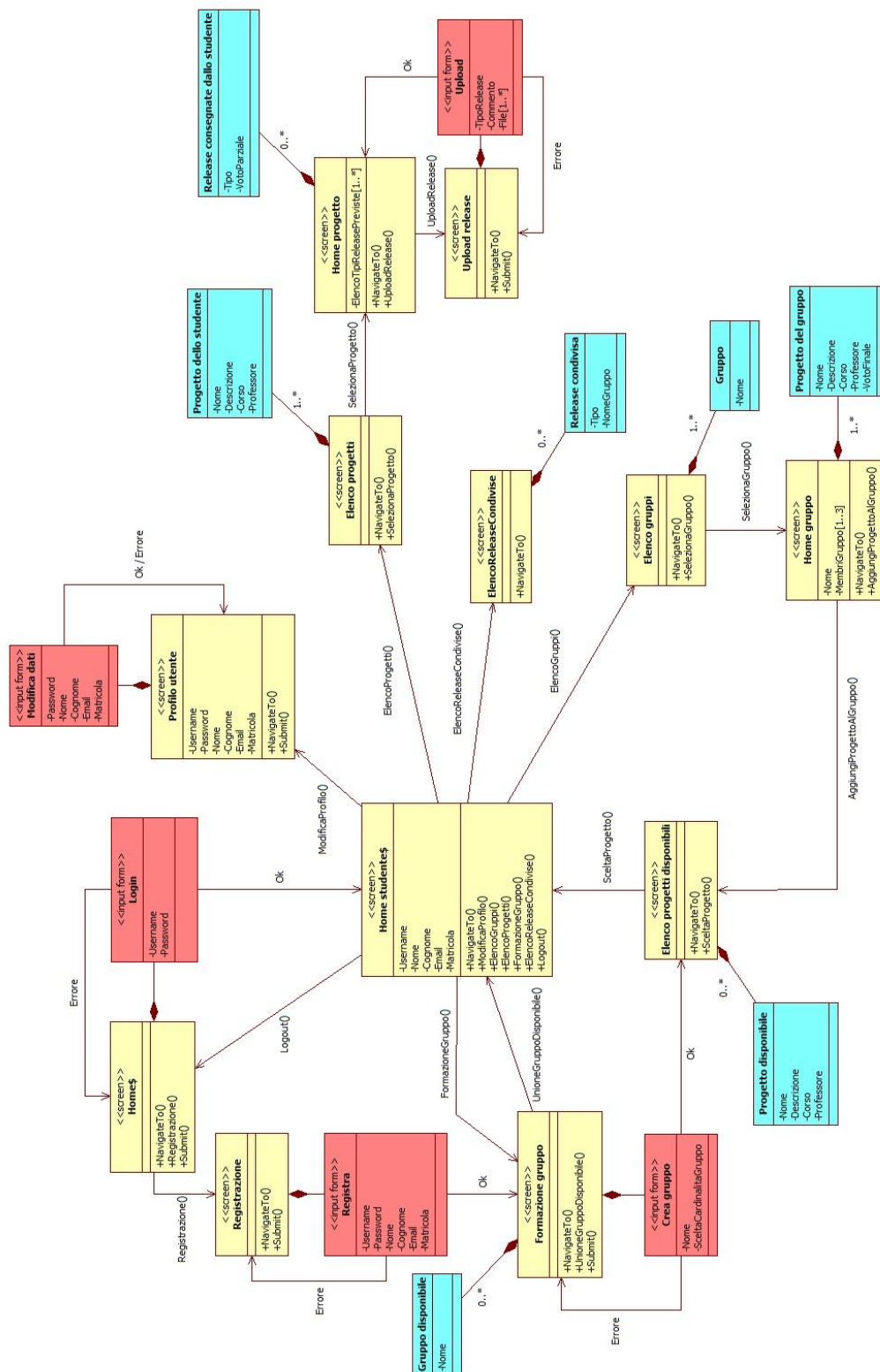


Figura 2.1 – Ux Model: Studente

## 2.1.2 Professore

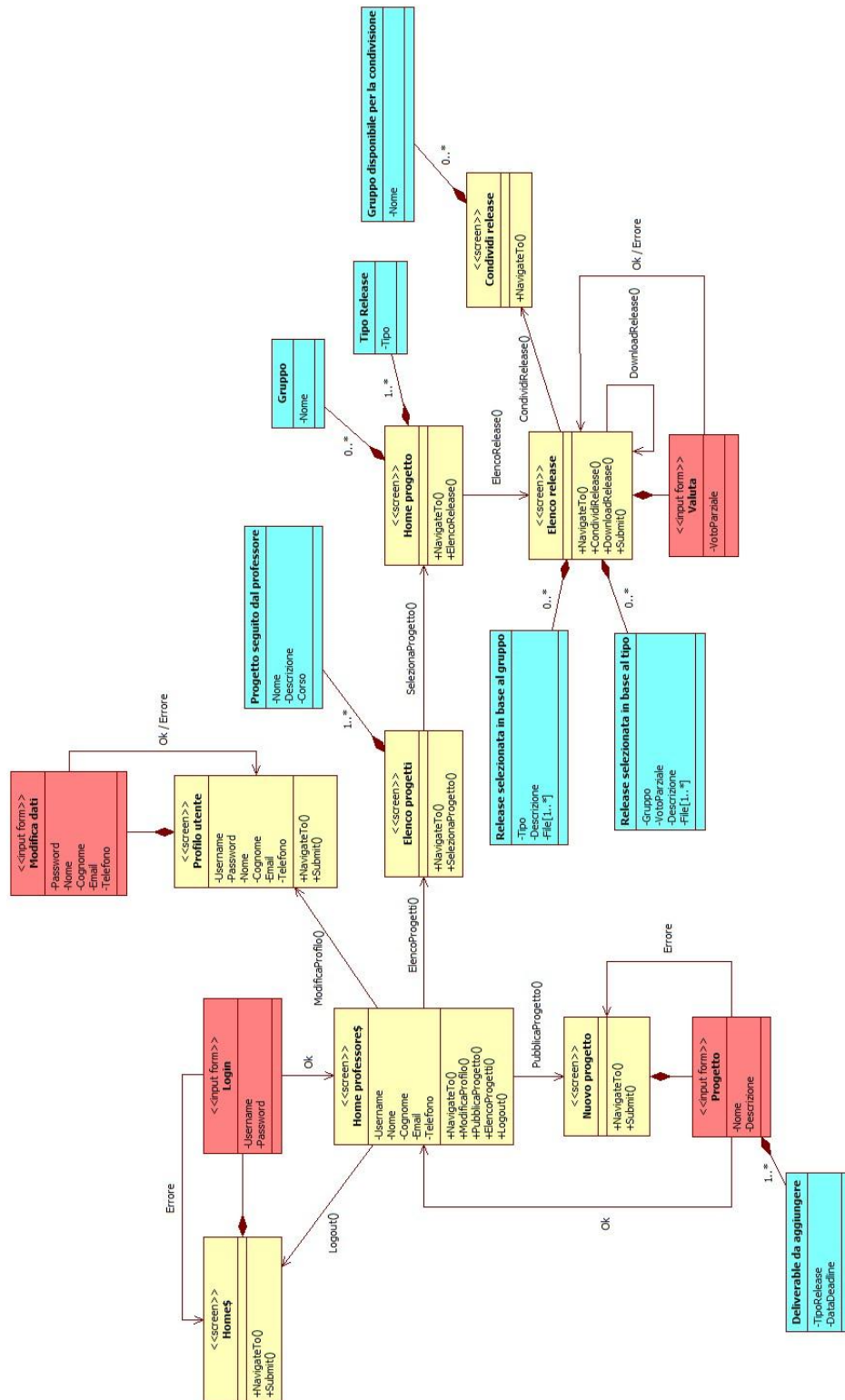


Figura 2.2 – Ux Model: Professore

### 2.1.3 Amministratore

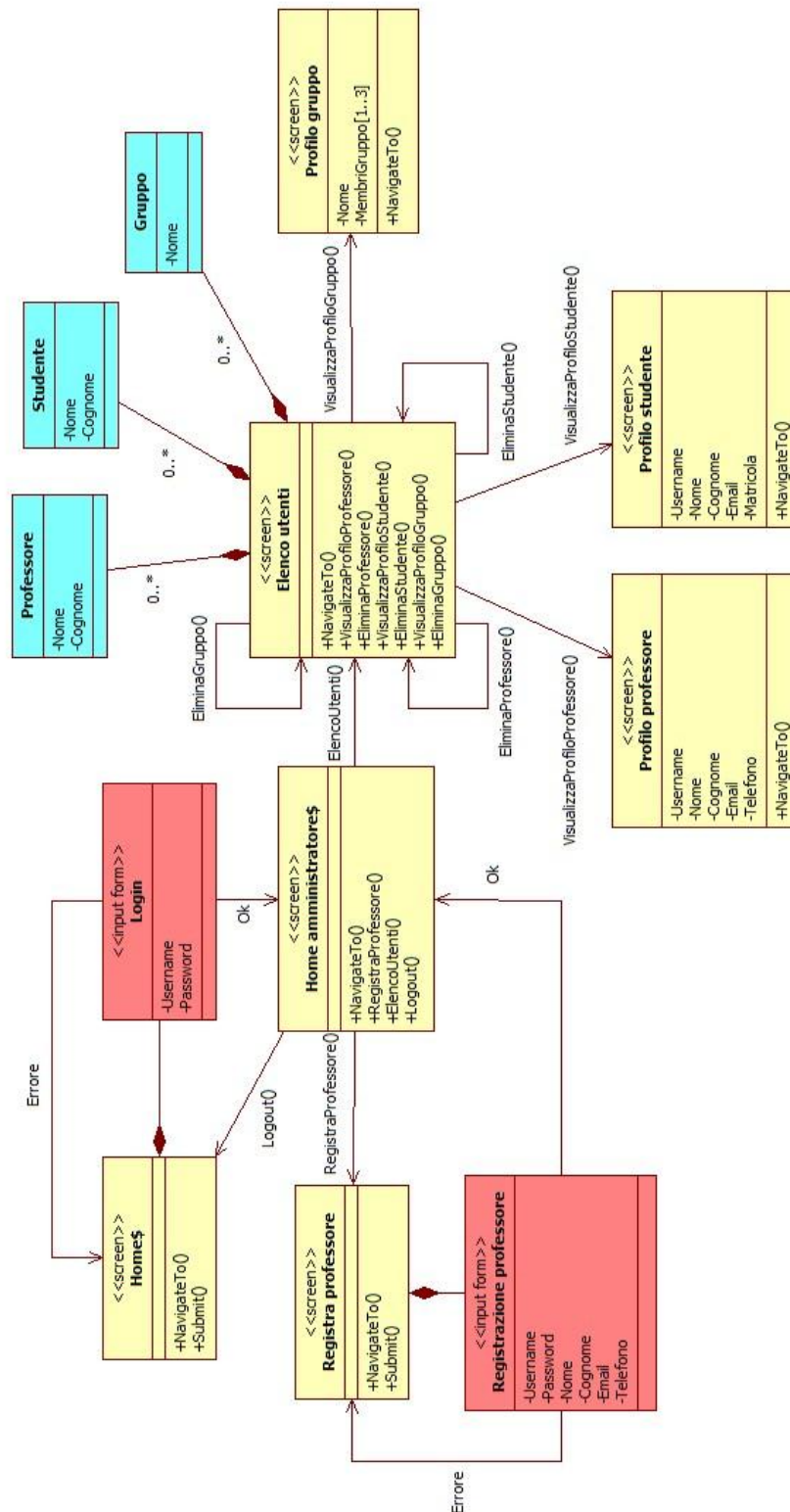


Figura 2.3 – Ux Model: Amministratore

## 2.2 Diagrammi di analisi

Per illustrare le tipologie dei singoli componenti (Boundary, Control, Entity) e le loro relazioni, di seguito verranno riportati i diagramma di analisi organizzati in relazione al tipo di utente.

### 2.2.1 Studente

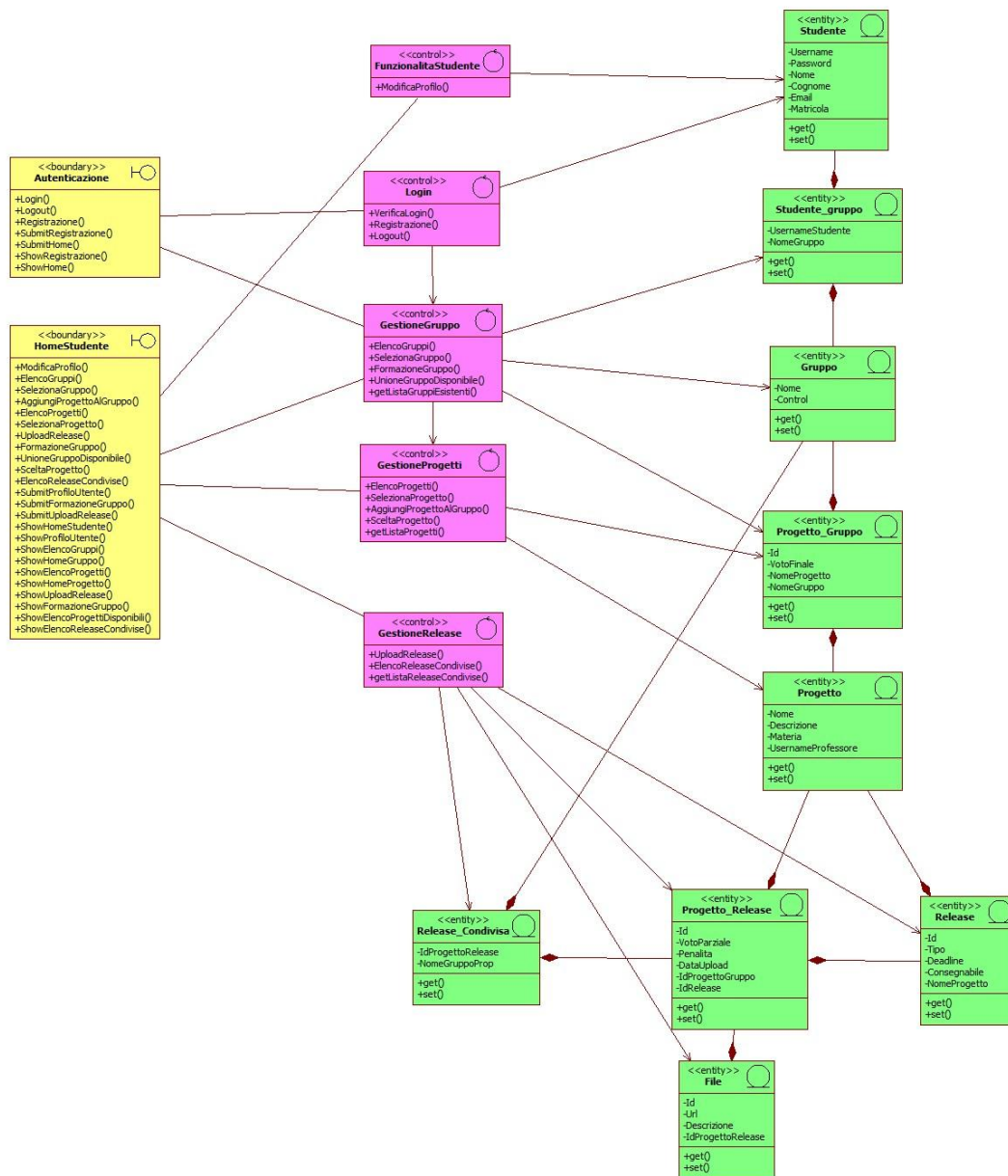


Figura 2.4 – BCE Model: Studente

## 2.2.2 Professore

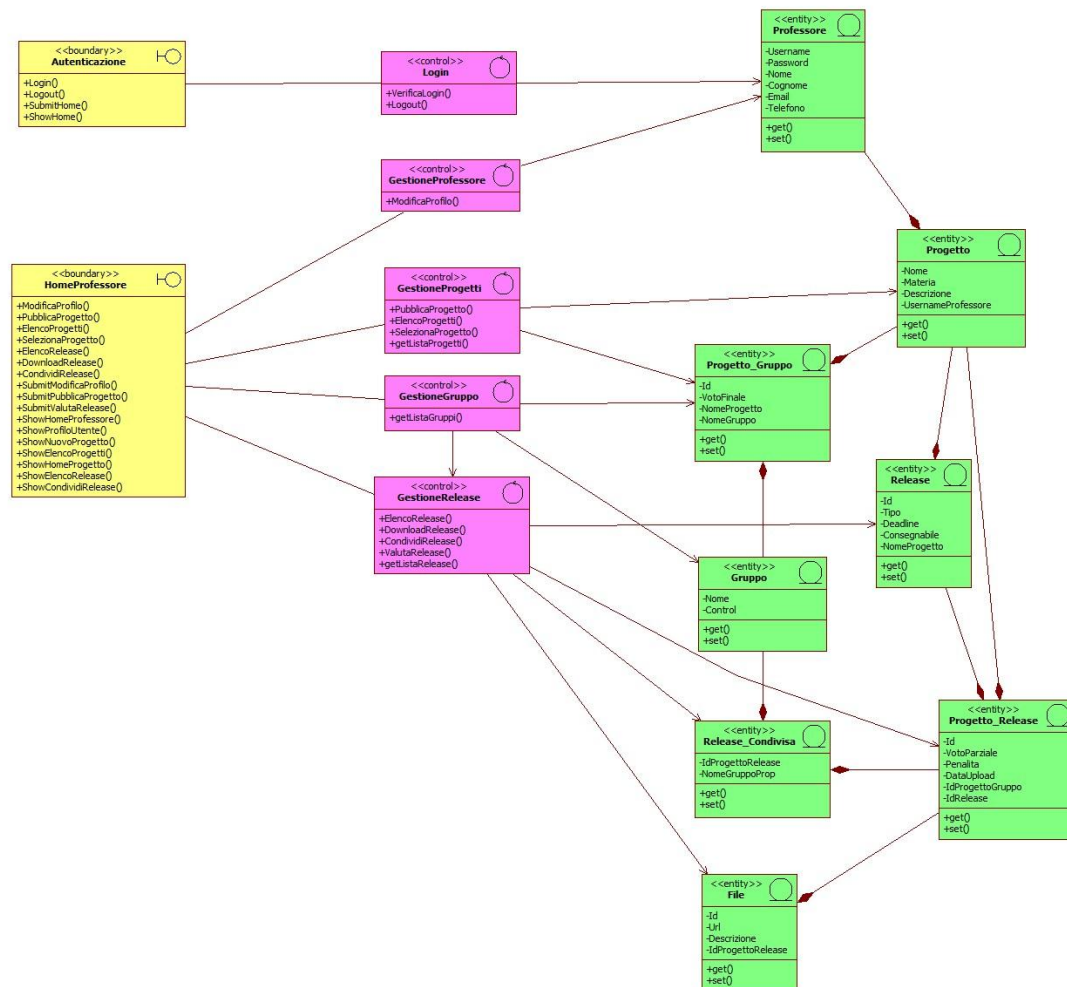


Figura 2.5 – BCE Model: Professore



## 2.2.3 Amministratore

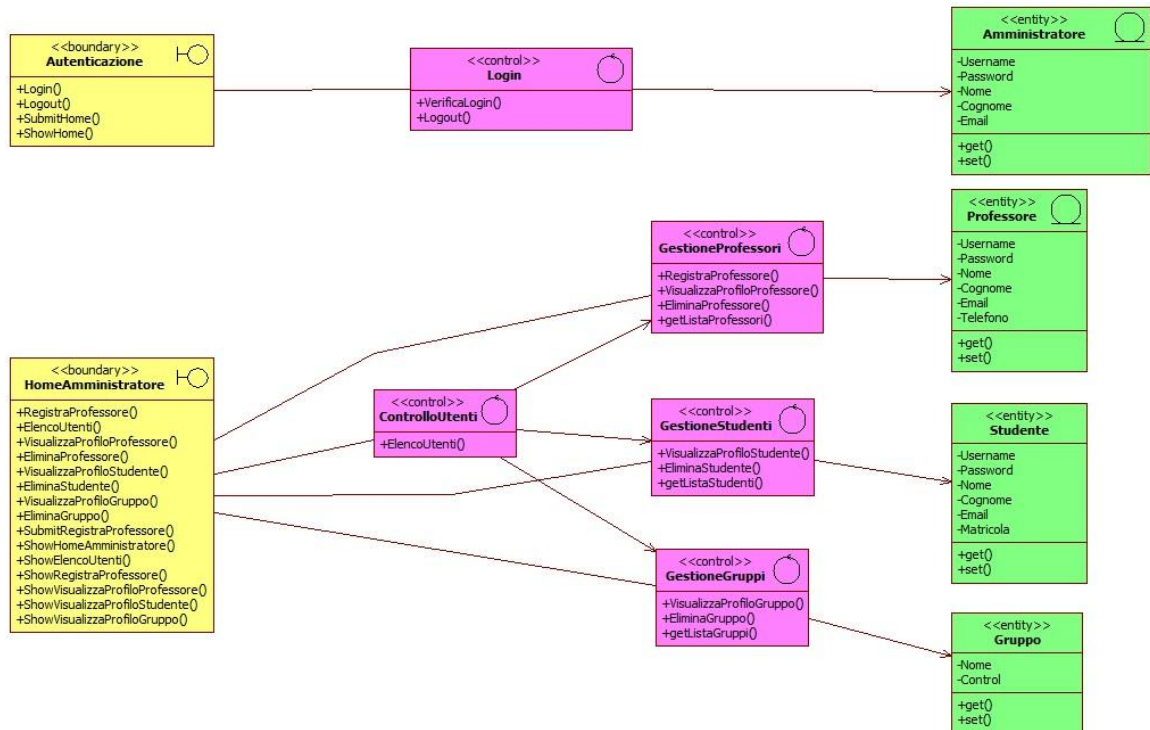


Figura 2.6 – BCE Model: Amministratore

## 2.3 Diagrammi di dettaglio

Per entrare maggiormente nel dettaglio sono stati esplosi i diagrammi di analisi, mettendo in evidenza i componenti dell'architettura JEE. Gli oggetti di tipo Boundary sono diventate pagine HTML/JSP e Servlet; gli oggetti di tipo Control hanno assunto la forma di EJB Session e infine gli oggetti di tipo Entity si sono trasformati in EJB Entity.

### 2.3.1 Login

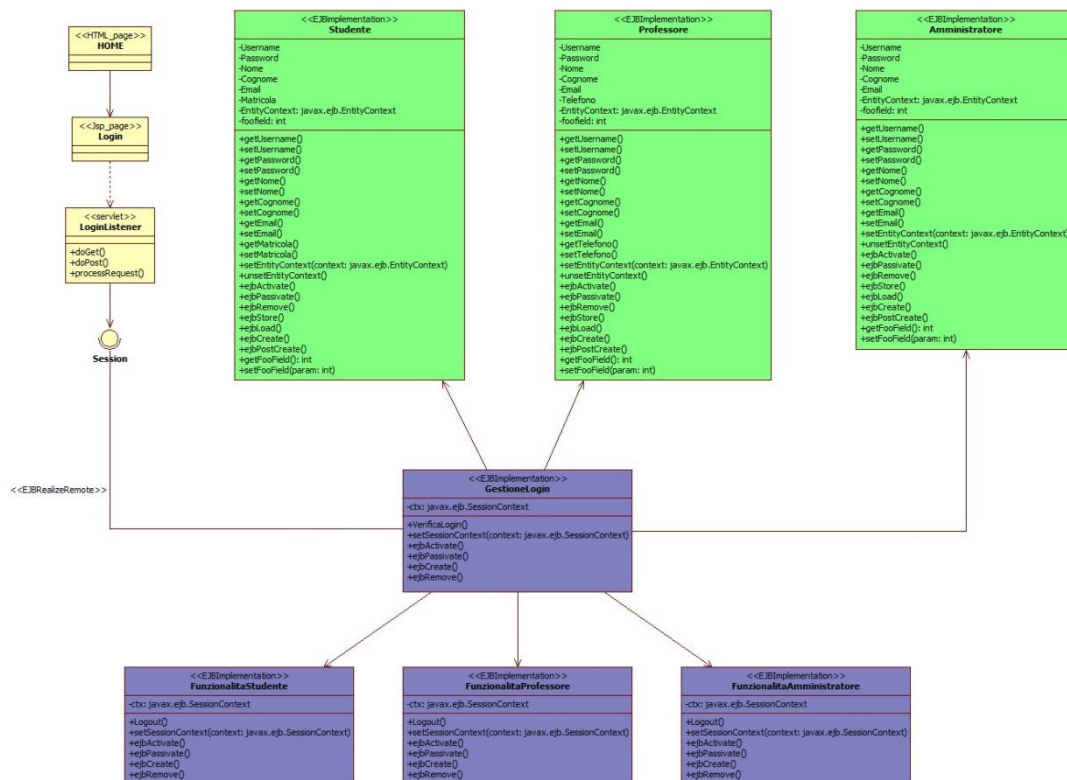


Figura 2.7 – JSP Model: Login

### 2.3.2 Studente

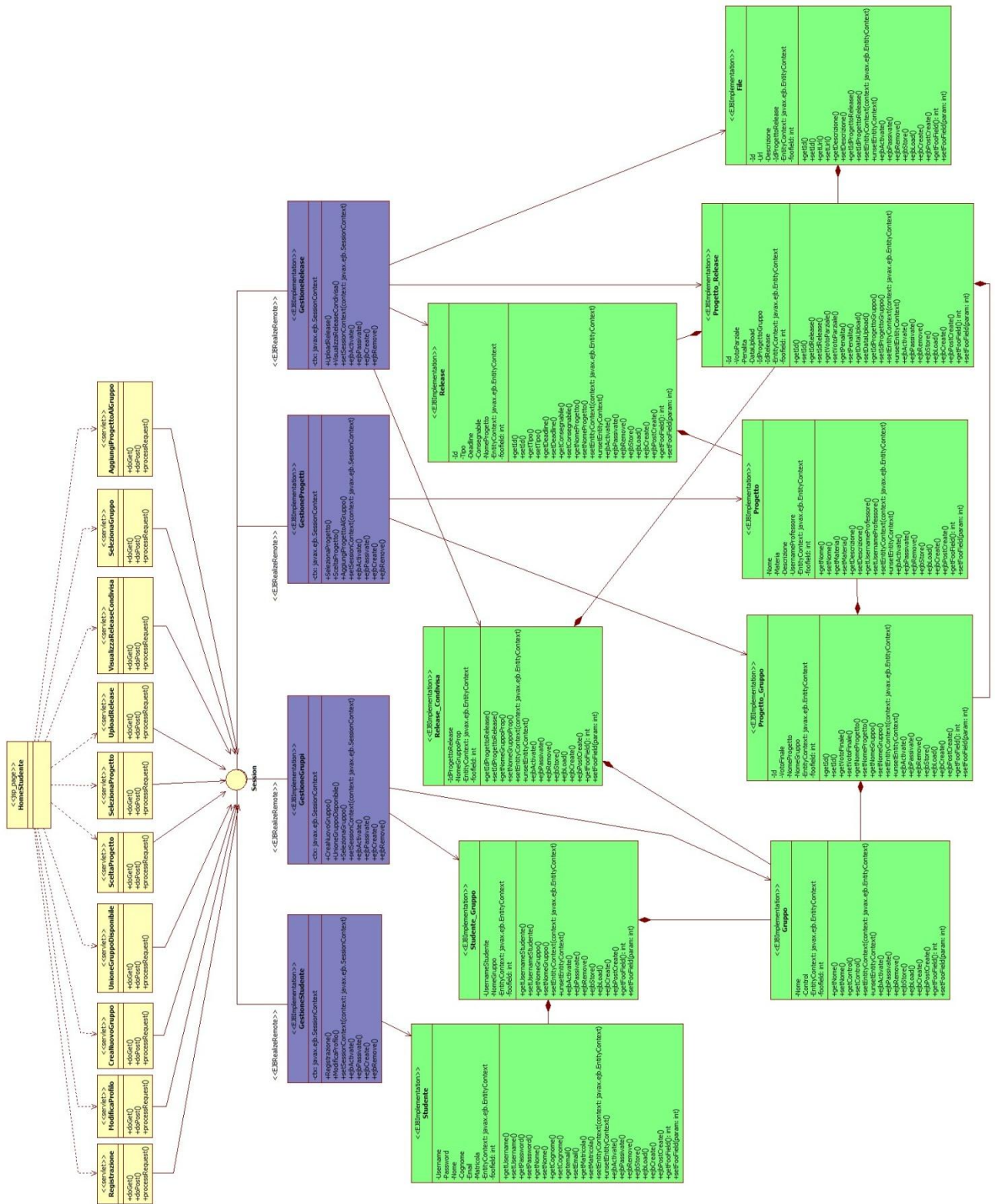


Figura 2.8 – JSP Model: Studente

### 2.3.3 Professore

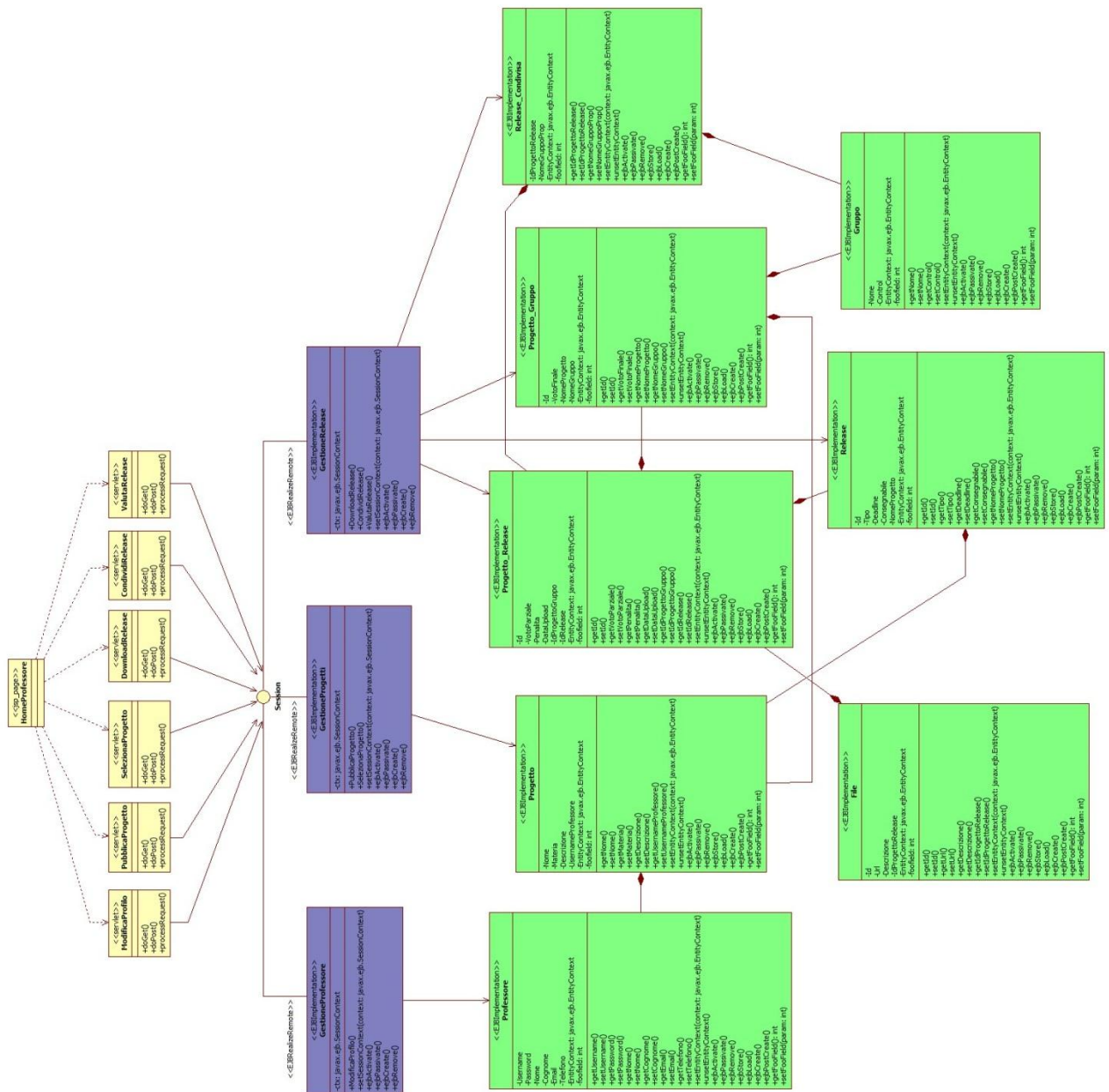


Figura 2.9 –JSP Model: Professore

## 2.3.4 Amministratore

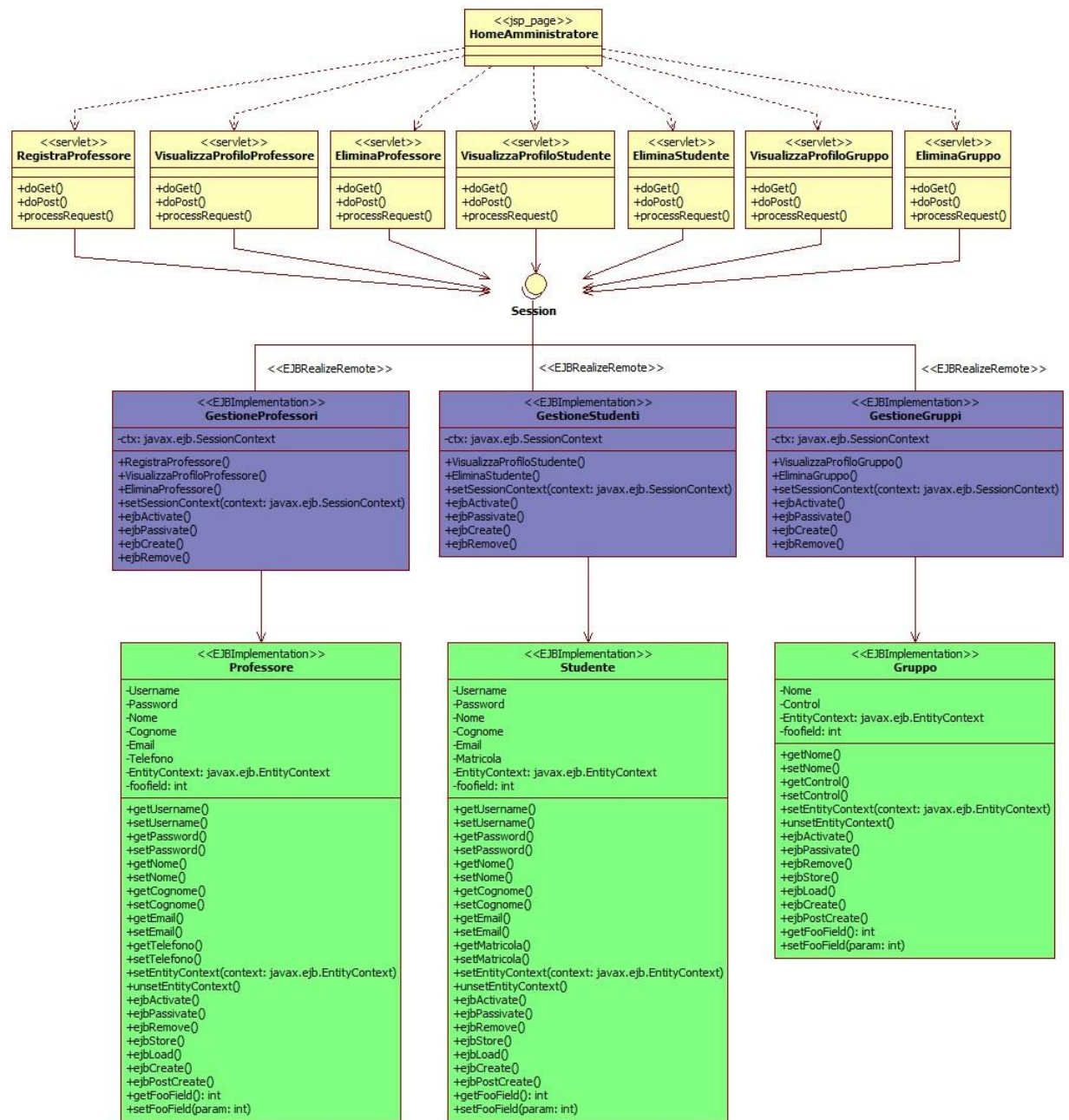


Figura 2.10 – JSP Model: Amministratore





## 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 MySQL

MySQL, definito Oracle MySQL, è un Relational Database Management System composto da un client con interfaccia a caratteri e un server, entrambi disponibili sia per sistemi Unix che per sistemi Windows.

### A.3 Microsoft Visio 2010

Microsoft Visio 2010 è uno strumento utilizzato per la creazione semplificata di diagrammi grazie ai suoi efficaci elementi visivi.