



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Cognome	Nome	Matricola	Indirizzo e-mail istituzionale
Cesarano	Agostino	N86003587	ag.cesarano@studenti.unina.it
Fortino	Alessandro	N86003884	ale.fortino@studenti.unina.it

1 Febbraio 2022

Sommario

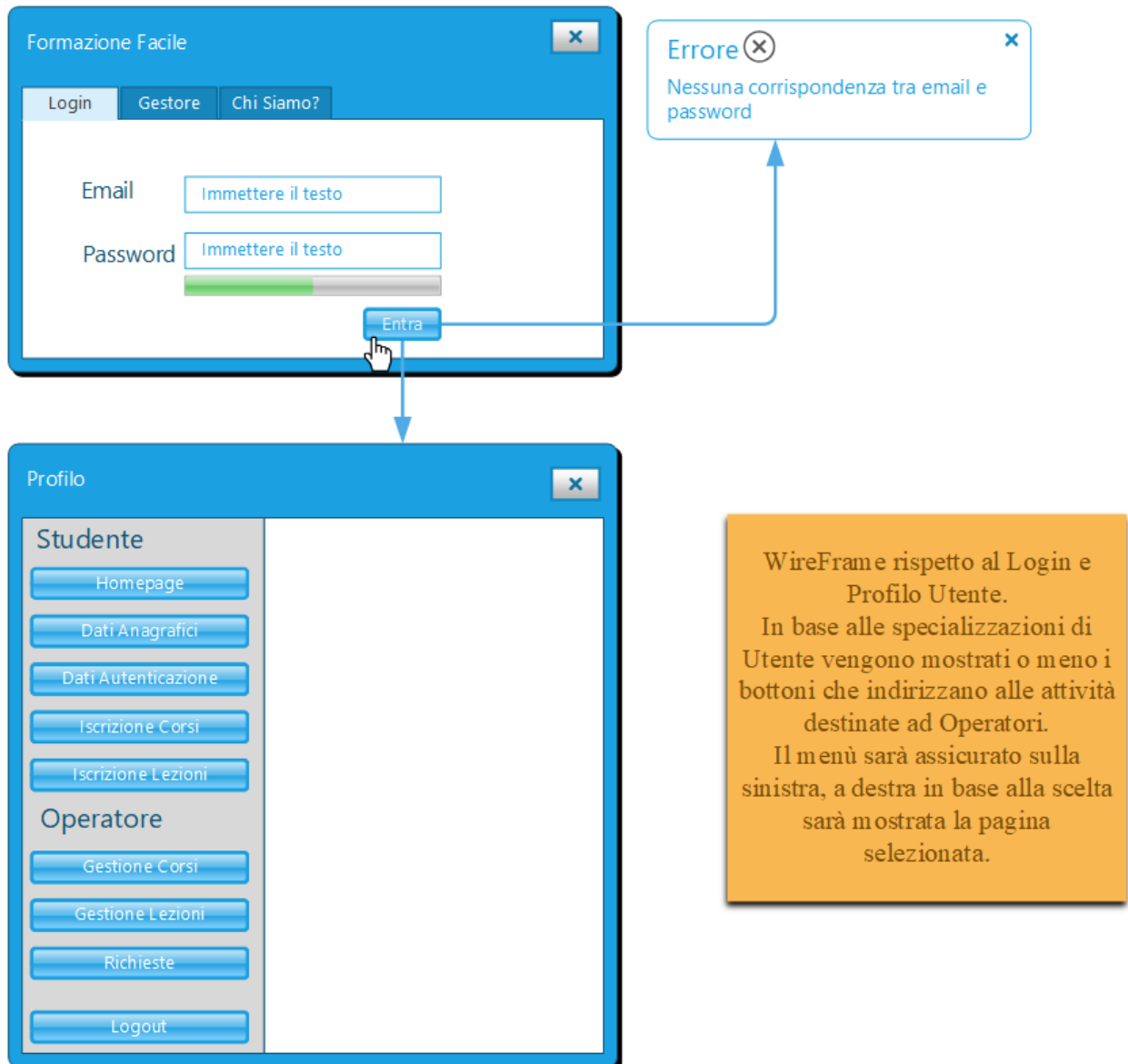
1. Analisi del problema	3
1.1 WireFrame.....	3
1.2 Tecnica di implementazione.....	5
1.3 Individuazione degli Oggetti.....	6
2. Stesura del Codice.....	7
2.1 Le Classi DAO	8
2.1.1 UML.....	9
2.2 Le Classi Controller	9
2.2.1 UML.....	10
2.3 Dipendenze	12
3. Diagramma delle Classi	13
4. Sequence Diagram	14

1. Analisi del problema

Dopo un accurata analisi della traccia, si è proseguito con l'analizzare il problema posto ricercando una soluzione Software che sia adeguata al problema stesso.

In primis si è ragionato ad un implementazione grafica **wireframe** che sia semplice e intuitiva, l'obiettivo prefissato è stato di mantenere un interfaccia minimal che quindi presenti pochi colori e che presenti meno Stage / Frame possibili.

1.1 WireFrame

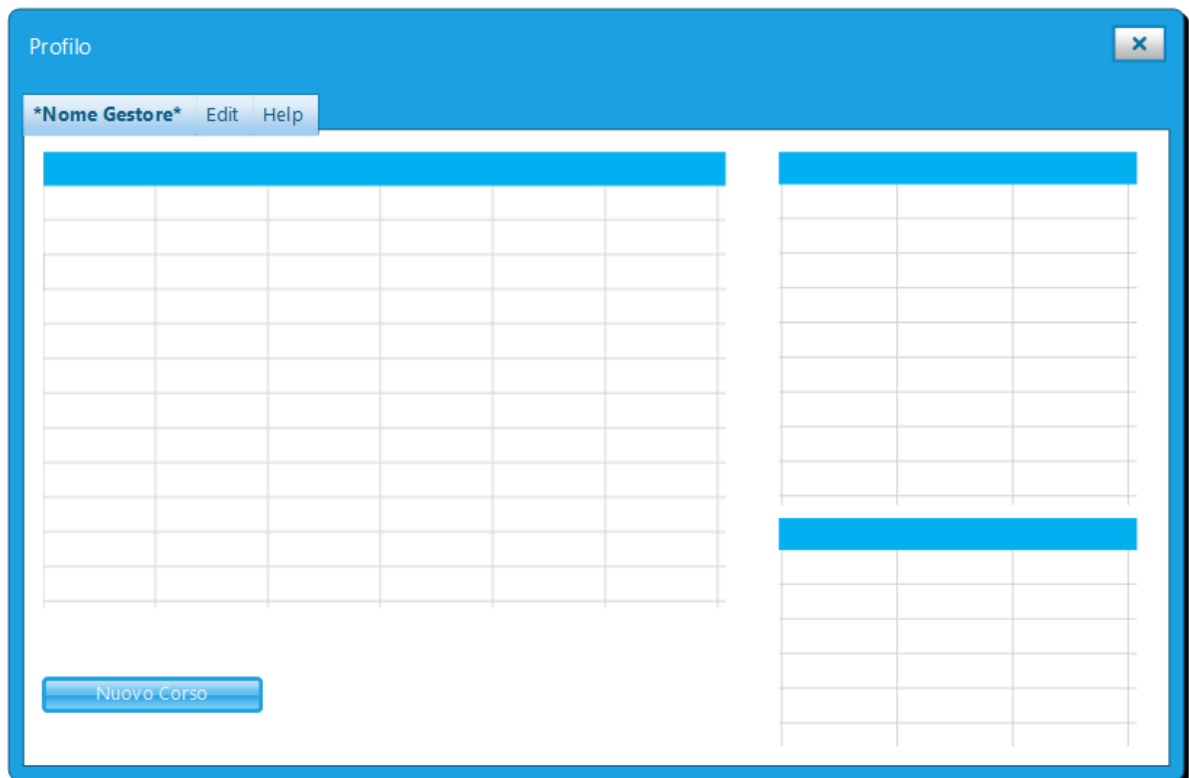


Le pagine visualizzate a destra sono composte per lo più da TableView che permettono l'interazione tramite click destri sulle righe che ci interessano.

La pagina principale, tramite un Tab Pane permette di differenziare il Login destinato agli Utenti rispetto a quello destinato ai Gestori di Corsi.

Effettuando il login tramite il tab riservato ai Gestori si accede al Profilo Gestore.

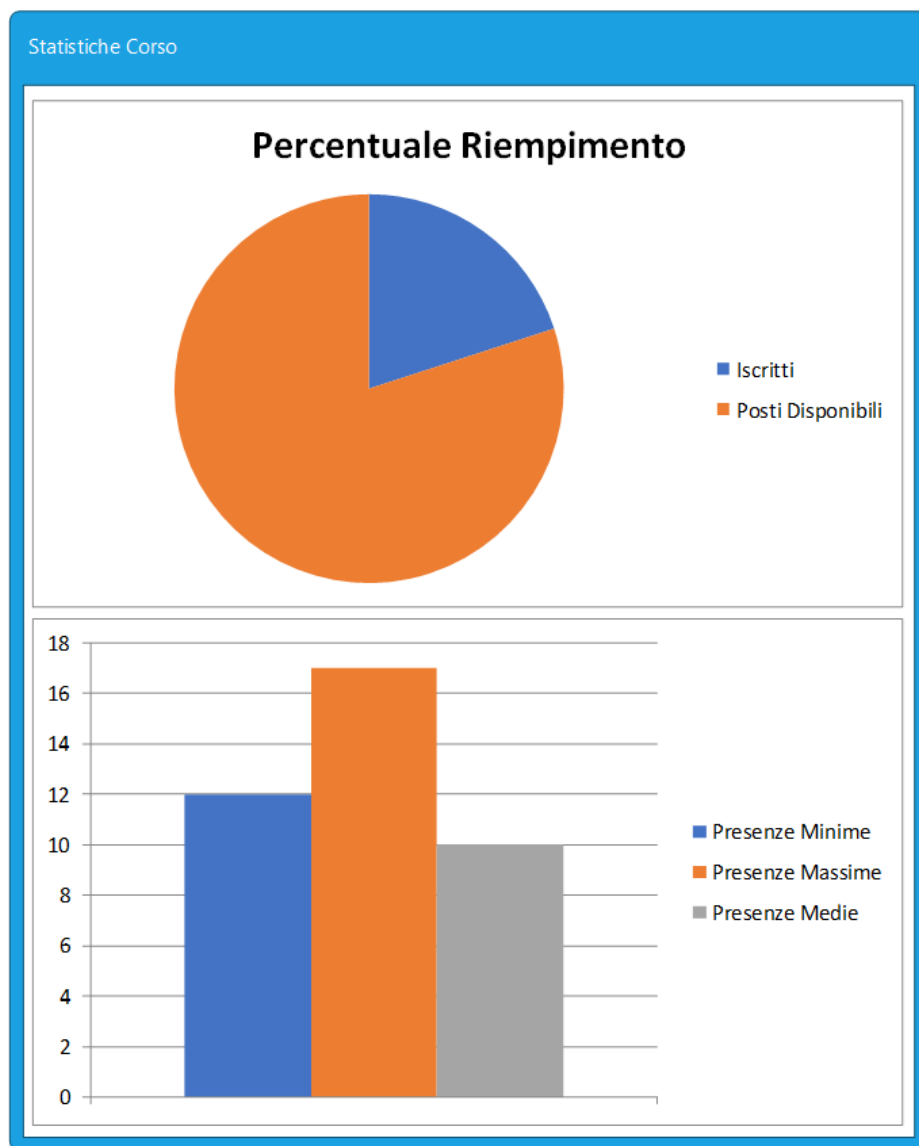
WireFrame Profilo Gestore



A differenza del Profilo Utente il Profilo Gestore tramite tre TableView mostra i dettagli su **Corsi, Operatori e Iscritti** dei Corsi erogati.

Come nel Profilo Utente si può interagire con ogni tabella tramite click destri sulle righe.

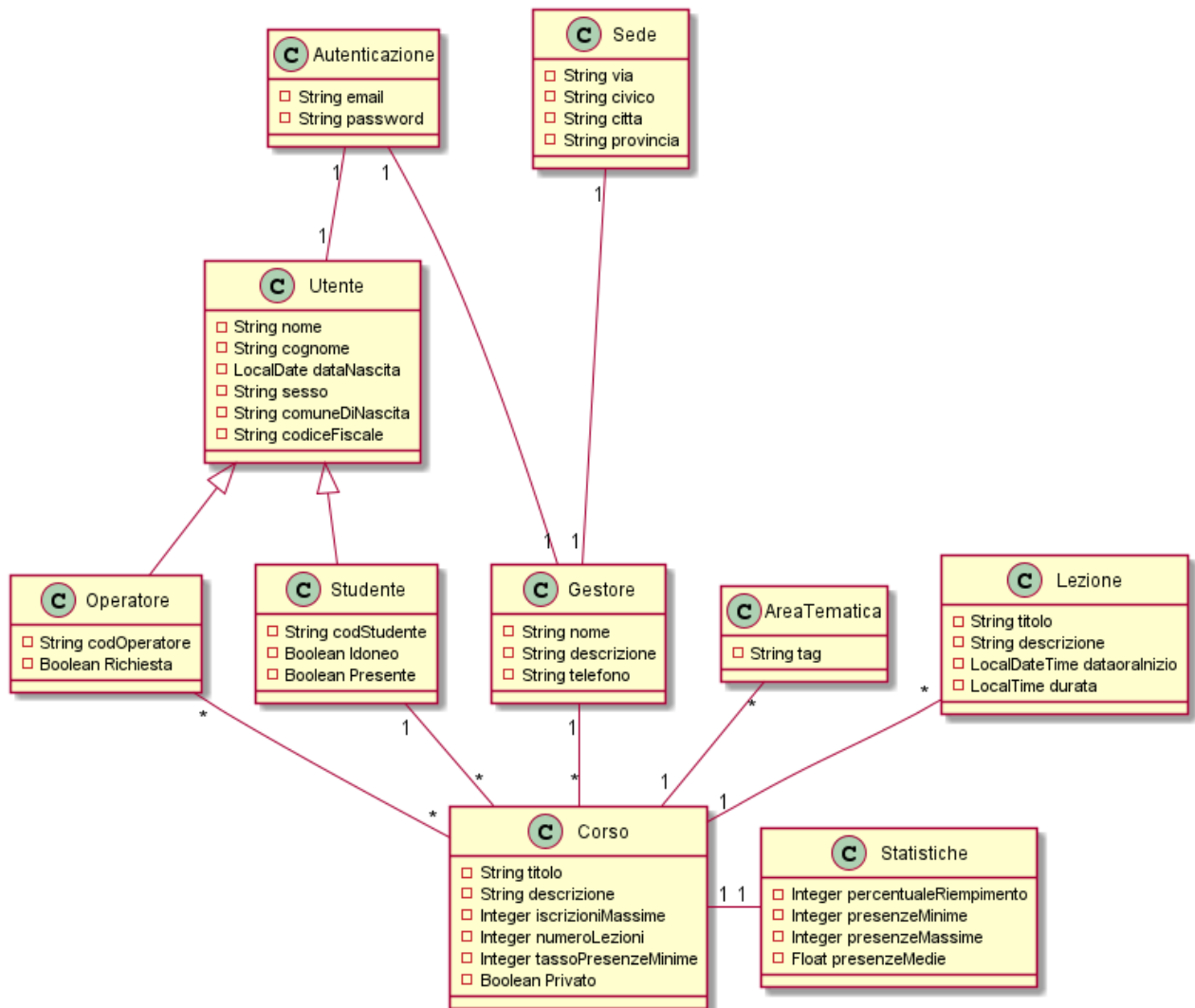
Gli **Operatori** e i **Gestori** hanno accesso alla pagina che permette la visualizzazione delle **Statistiche** del Corso, che vengono mostrate tramite Grafici.



1.2 Tecnica di implementazione

In seguito abbiamo pensato come implementare l'idea sviluppata precedentemente in **wireframe** tramite **Java**, valutando tra **Swing** e **Java FX**, la scelta è ricaduta sull'ultimo poiché mette a disposizione la possibilità di mostrare i dati richiesti dalla traccia tramite Grafici, permette di avere un interfaccia decisamente più moderna e avendo precedenti conoscenze in **CSS** che è utilizzabile in **FX**.

1.3 Individuazione degli Oggetti

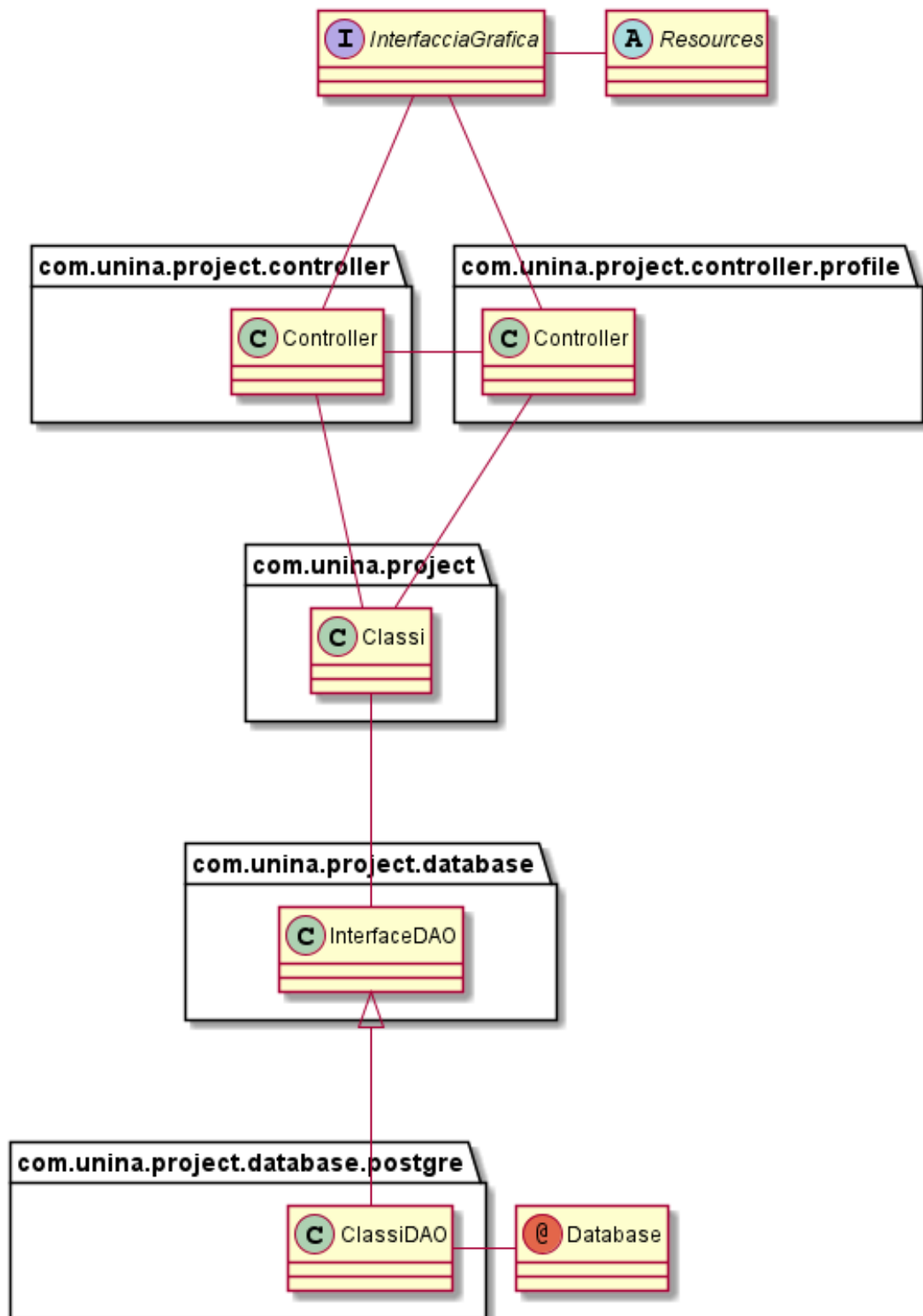


Individuati ciò che saranno gli Oggetti del Programma possiamo procedere con la stesura del codice.

I dettagli di ogni Classe sono riportati nel paragrafo dedicato alle Carte di Class Responsibility Collaboration (**CRC Cards**) .

2. Stesura del Codice

Avendo le idee chiare su come implementare l'idea tramite Software si è iniziato suddividendo il programma in vari **package**.

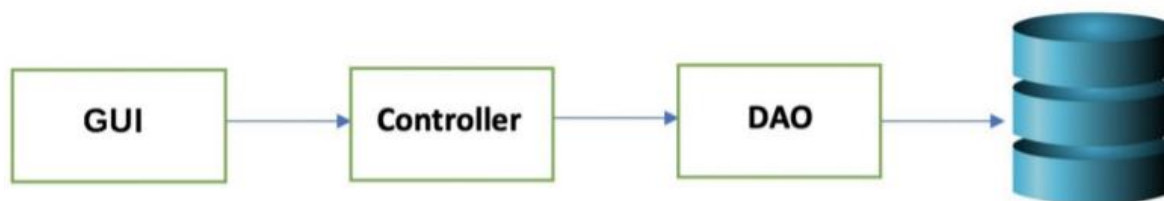


Ogni **package** comunica con l'altro in modo bidirezionale, ogni classe richiama elementi di altri **package**. La suddivisione in più pacchetti è utile per suddividere le varie parti del Programma.

<i>Package</i>	<i>Funzione</i>
com.unina.project	contiene gli Oggetti/Tipi definiti nel Programma.
com.unina.project.controller e com.unina.project.controller.profile	contengono tutte le classi Controller che interagiscono con l'interfaccia grafica e con gli fxml contenuti nelle resources .
com.unina.project.database	contiene le interfacce dei DAO relativi al Database.
com.unina.project.database.postgre	contiene le classi DAO specifiche postgre che implementano le interfacce.

2.1 Le Classi DAO

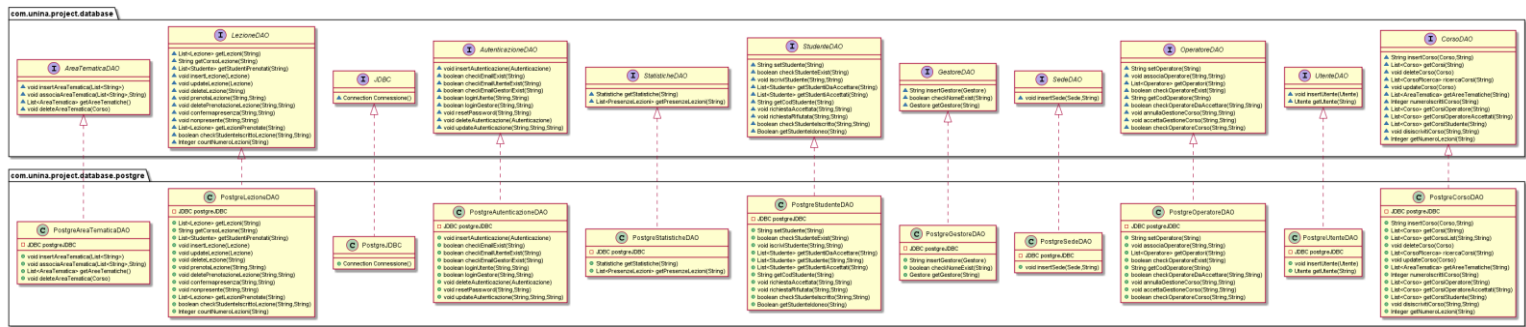
Le Classi **Data Access Object** sono state aggiunte per permettere al resto del Programma di essere indipendente dal Database. Le Query sono contenute esclusivamente al loro interno per permettere al Software nel caso in cui si cambiasse Database di modificare esclusivamente ciò che sono le classi che implementano le **Interface**.



Le implementazioni delle **interface DAO** a loro volta verranno interfacciate con i **Controller**, che avranno il compito di fare da intermediari con le interfacce grafiche, e quindi far **isolare** completamente le interfacce grafiche dal mondo della base dati.

2.1.1 UML

UML dei package che contengono Interfacce e Classi DAO



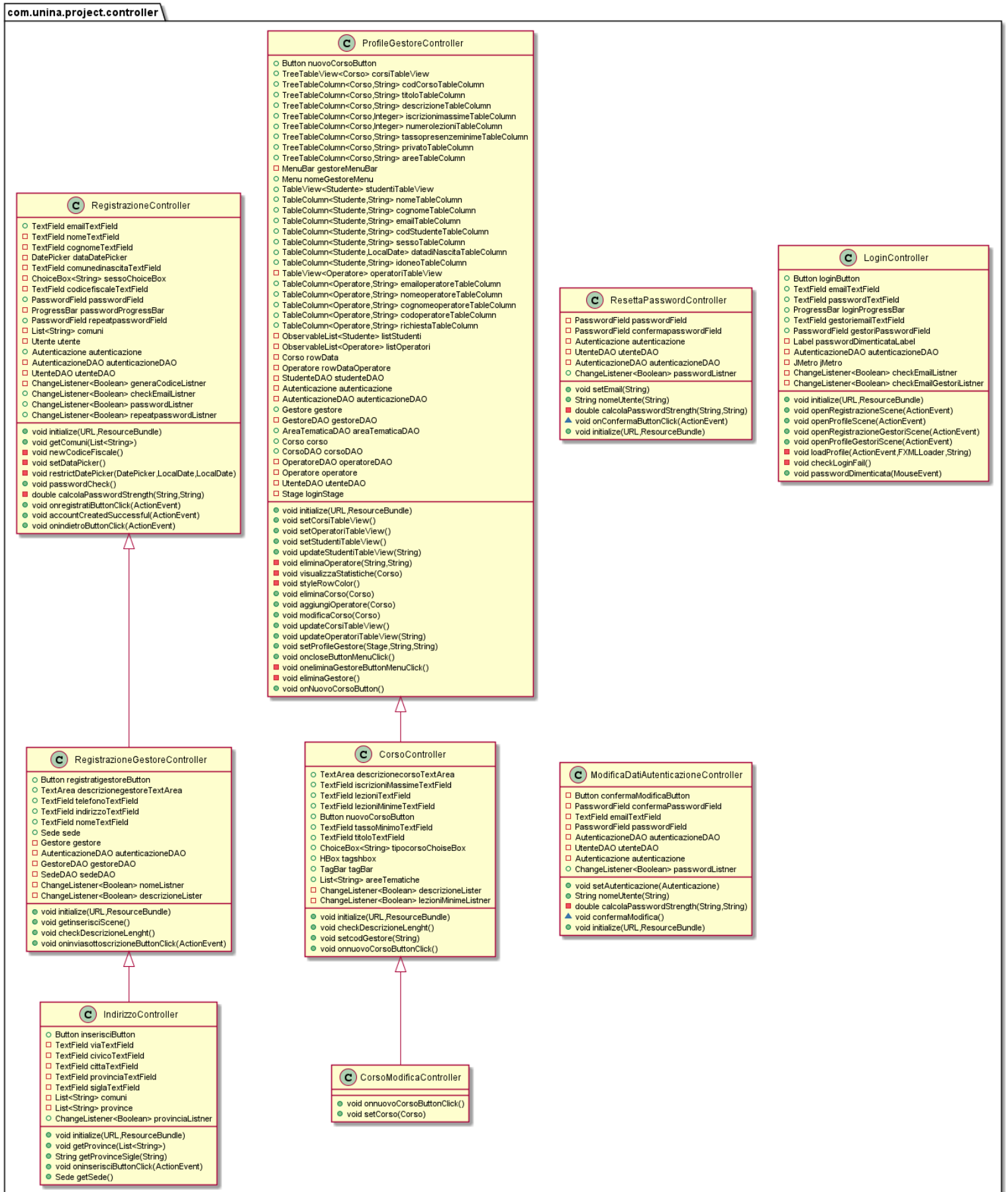
Si consiglia di fare click sopra l'immagine che apre il file PNG per una visione migliore

2.2 Le Classi Controller

Le classi Controller sono le classi che gestiscono l'interfaccia grafica. Ogni **Controller** è associato ad un **fxml**, contenuto nelle **resources**, che definisce tutti gli elementi grafici dell'interfaccia. Ad ogni elemento può essere associato un **id**, che definito nel Controller, permette la gestione/personalizzazione dell'elemento.

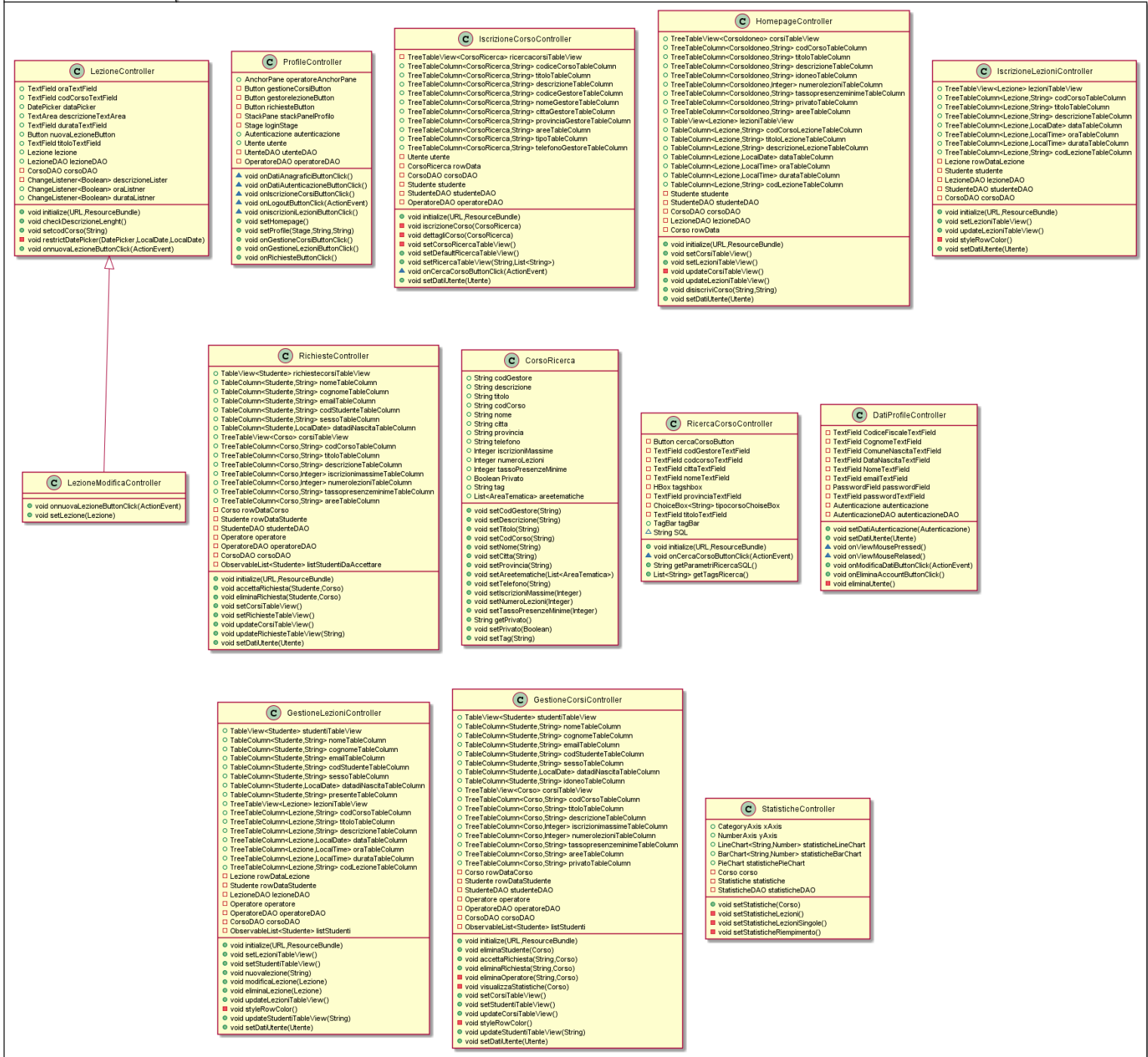
I Controller sono distribuiti in **due package**, un package è dedicato ai Controller del Profilo Utente, l'altro ai restanti.

2.2.1 UML



SISTEMA DI GESTIONE PER CORSI DI FORMAZIONE [OObD2122]

com.unina.project.controller.profile

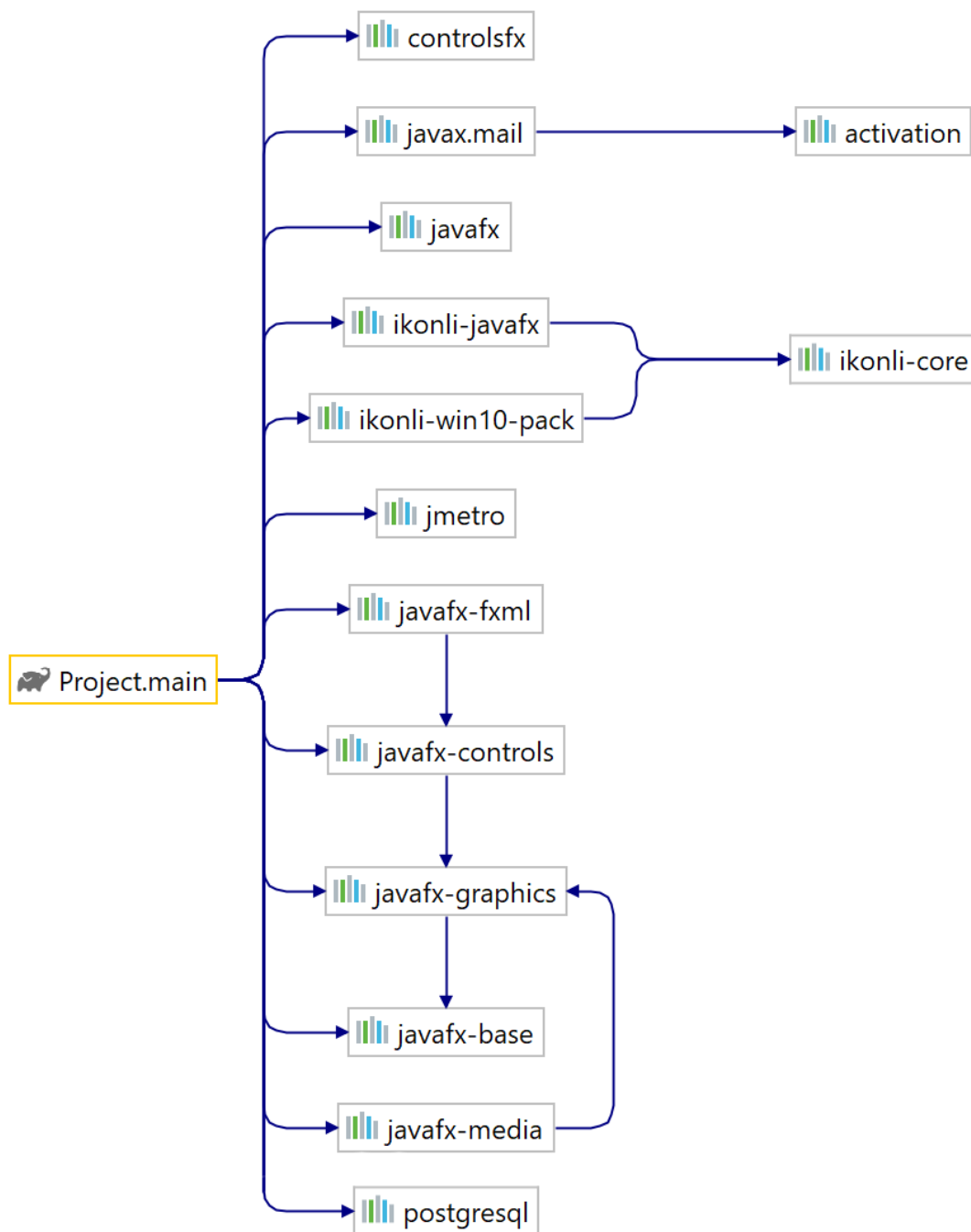


Si consiglia di fare click sopra l'immagine che apre il file PNG per una visione migliore

2.3 Dipendenze

Il funzionamento del programma dipende da dei **moduli terzi** che permettono l'uso di funzioni non implementate nativamente nel **JDK**.

Le dipendenze sono gestite da **Gradle** che è uno strumento di automazione della compilazione (**build automation tool**).



<i>Moduli</i>	<i>Funzione</i>
JavaFX	Aggiunge tutte le funzionalità grafiche di JavaFX, anche se sviluppato da Oracle non è contenuto nel JDK.
ControlsFX	Permette di aggiungere elementi grafici avanzati e funzionalità extra. Nel progetto è stato usato per le TextField con Suggerimenti.
JavaX.Mail	Permette di interfacciarsi con il mondo mail tramite Java. Nel progetto si inviano codici di verifica durante la registrazione.
Ikonli	Permette di aggiungere Icone all'interno degli FXML senza ricorrere ad immagini.
JMetro	Tema Grafico ispirato all'interfaccia Windows.
PostgreSQL	Aggiunge le funzionalità di iterazione Java-Database JDBC (Java DataBase Connectivity).

[Consulta la JavaDoc per visualizzare i dettagli.](#)

3. Diagramma delle Classi

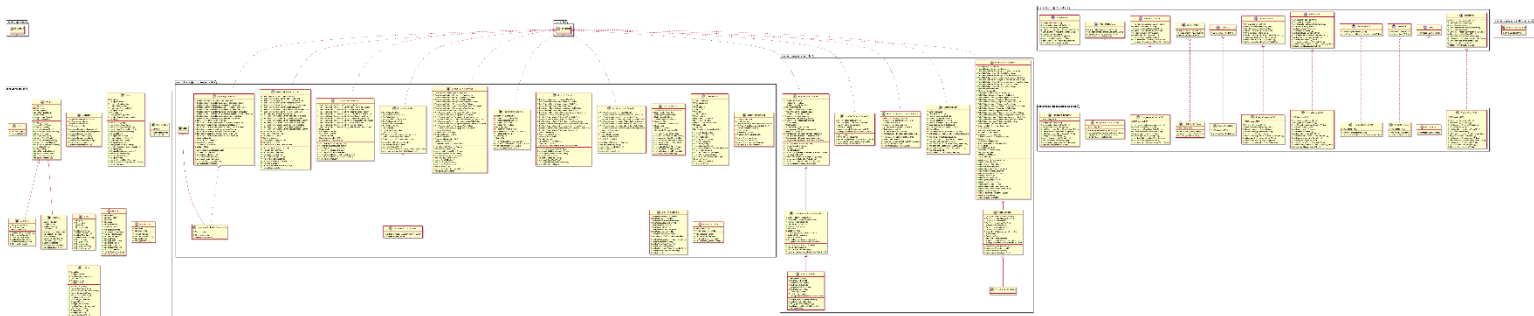
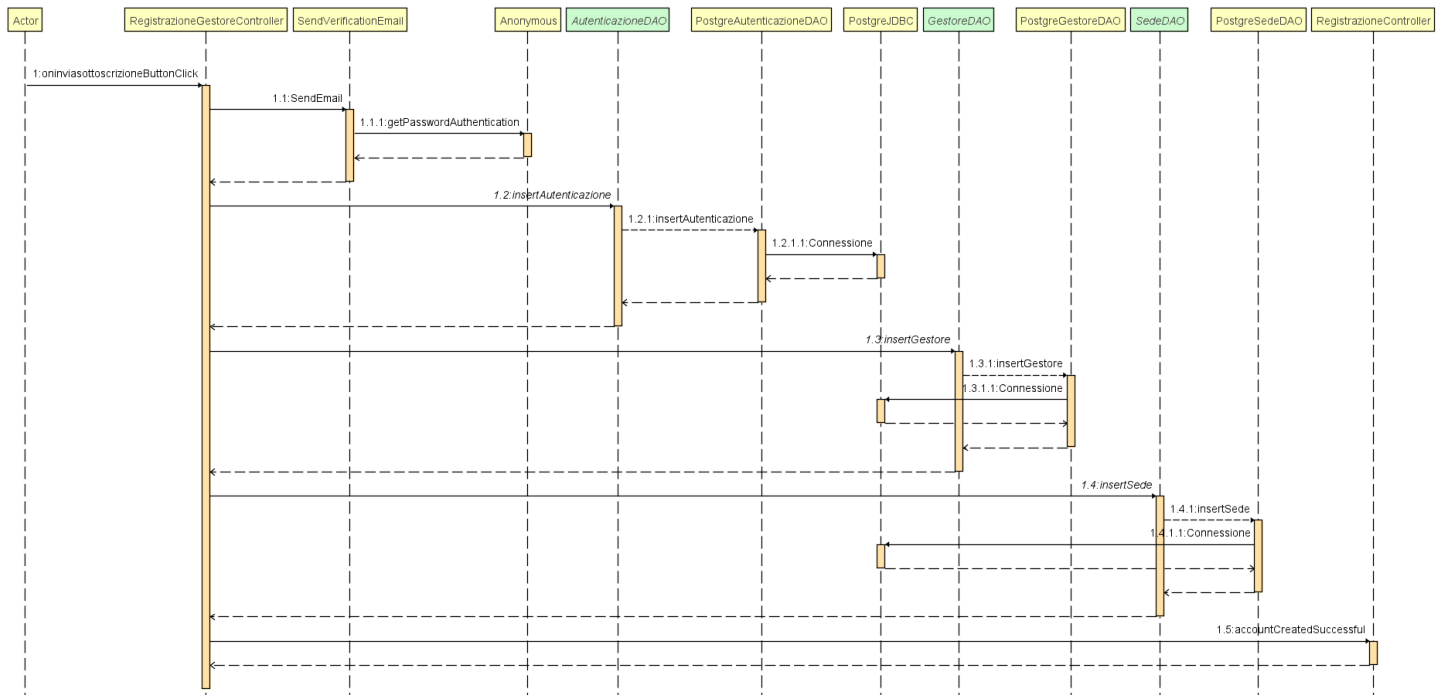


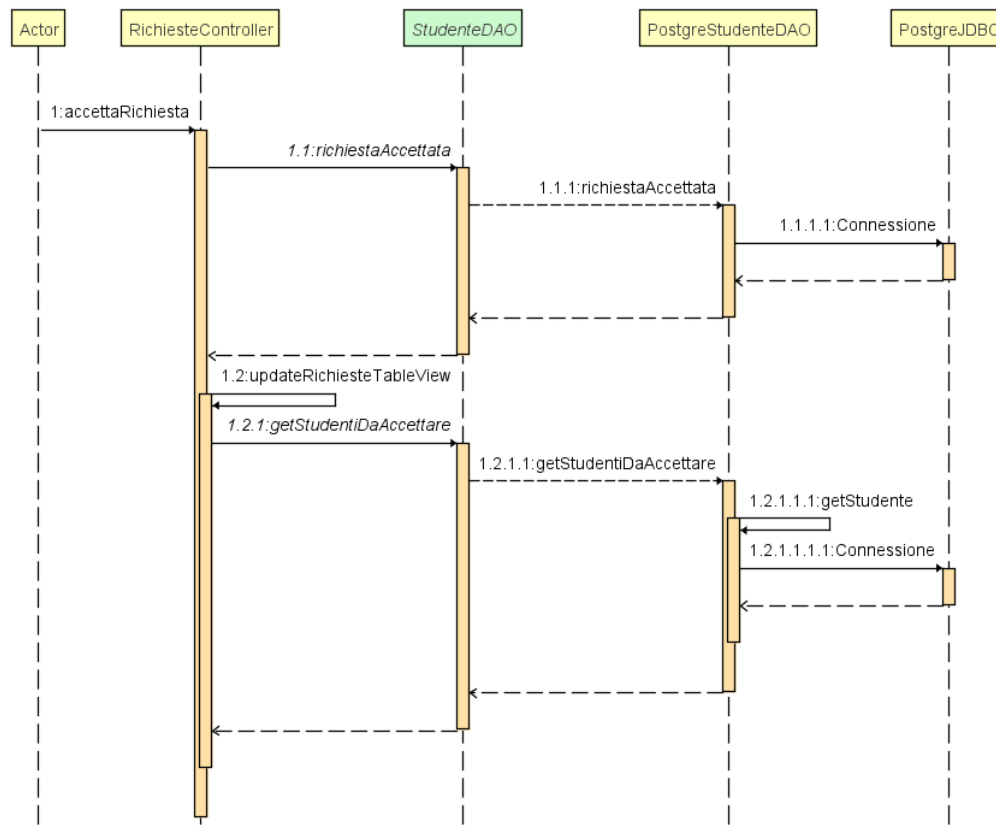
Diagramma delle Classi Completo dell'intero progetto, a causa della dimensione elevata, si consiglia la visione esterna.

[Consulta il PDF dedicato.](#)

4. Sequence Diagram



Sequence Diagram del metodo *oninviasottoscrizioneButtonClick* del Controller *RegistratoreGestoreController*



Sequence Diagram del metodo *accettaRichiesta* del Controller *RichiesteController*

5. CRC Card

Autenticazione	
Description: Login Information	
Attributes:	
Name	Description
email	
password	
Responsibilities:	
Name	Collaborator
Contiene le informazioni di Login di Utenti e Gestori	

Utente	
Sub Classes: Operatore, Studente	
Description: Utente Information	
Attributes:	
Name	Description
nome	
cognome	
dataNascita	
sex	
comuneDiNascita	
codiceFiscale	
email	Riferimento BD
Responsibilities:	
Name	Collaborator
Contiene le informazioni dell'Utente	Autenticazione

Gestore	
Description: Gestore Information	
Attributes:	
Name	Description
nome	
descrizione	
telefono	
codGestore	Identificativo Gestore BD
corsi	List di Corso
email	Riferimento BD
Responsibilities:	
Name	Collaborator
Contiene le informazioni del Gestore	Autenticazione
	Corso

Studente	
Super Classes: Utente	
Description: Studente Information	
Attributes:	
Name	Description
Idoneo	
Presente	
codStudente	Identificativo Studente BD
corsi	List di Corso
Responsibilities:	
Name	Collaborator
Contiene le informazioni dello Studente	Corso

Operatore	
Super Classes: Utente	
Description: Operatore Information	
Attributes:	
Name	Description
Richiesta	Stato della richiesta di gestione
codOperatore	Identificativo Operatore BD
corsi	List di Corso
Responsibilities:	
Name	Collaborator
Contiene le informazioni dell'Operatore	Corso

Corso	
Description: Corso Information	
Attributes:	
Name	Description
titolo	
descrizione	
iscrizioniMassime	
numeroLezioni	
tassoPresenzeMinime	
Privato	Tipo Corso Privato/Pubblico
codCorso	Identificativo Corso BD
operatori	List di Operatore
areetematiche	List di Area Tematica
Responsibilities:	
Name	Collaborator
Contiene le informazioni del Corso	Lezione
	Studente
	Operatore
	Area Tematica
	Statistiche

SISTEMA DI GESTIONE PER CORSI DI FORMAZIONE [O0BD2122]

Area Tematica	
Description: Area Tematica Information	
Attributes:	
Name	Description
tag	
Responsibilities:	
Name	Collaborator
Contiene le informazioni dell'areatematica	Corso

Lezione	
Description: Lezione Information	
Attributes:	
Name	Description
titolo	
descrizione	
dataorainizio	
durata	
codLezione	Identificativo Lezione BD
codCorso	Riferimento BD
Responsibilities:	
Name	Collaborator
Contiene le informazioni delle lezioni	Corso

Statistiche	
Description: Statistiche Information	
Attributes:	
Name	Description
percentualeRiempimento	
presenzeMinime	
presenzeMassime	
presenzeMedie	
Responsibilities:	
Name	Collaborator
Contiene le informazioni delle Statistiche	

ResettaPasswordController	
Description: Permette la modifica della password	
Attributes:	
Name	Description
lezioniMinimeListner	
confermapasswordField	
autenticazione	
utenteDAO	
autenticazioneDAO	
passwordListner	
Responsibilities:	
Name	Collaborator
Permette il reset e la modifica della password sia di un utente che di un gestore.	

CorsoModificaController	
Super Classes: CorsoController	
Description:	
Attributes:	
Name	Description
Responsibilities:	
Name	Collaborator
Permette la modifica dei corsi.	

CorsoController	
Super Classes: GestoreController	
Sub Classes: CorsoModificaController	
Description: Rappresenta i vari corsi erogati dai gestori	
Attributes:	
Name	Description
descrizionecorsoTextArea	
iscrizioniMassimeTextField	
lezioniTextField	
lezioniMinimeTextField	
nuovoCorsoButton	
tassoMinimoTextField	
titoloTextField	
tipocorsoChoiseBox	
tagshbox	
tagBar	
areeTematiche	
descrizioneLister	
lezioniMinimeListner	
Responsibilities:	
Name	Collaborator
Contiene le informazioni di tutti i corsi	

ModificaDatiAutenticazioneController	
Description: Permette la modifica dei dati	
Attributes:	
Name	Description
confermaModificaButton	
confermaPasswordField	
emailTextField	
passwordField	
autenticazioneDAO	
utenteDAO	
autenticazione	
passwordListner	
Responsibilities:	
Name	Collaborator
Permette la modifica dei dati di accesso per gli utenti.	

SISTEMA DI GESTIONE PER CORSI DI FORMAZIONE [O0BD2122]

GestoreController	
Sub Classes: CorsoController	
Description: Rappresenta la main page di tutti i gestori	
Attributes:	
Name	Description
nuovoCorsoButton	
corsiTableView	
codCorsoTableColumn	
titoloTableColumn	
descrizioneTableColumn	
iscrizioneemassimeTableColumn	
numerolezioniTableColumn	
tassopresenze minimeTableColumn	
privatoTableColumn	
areeTableColumn	
gestoreMenuBar	
nomeGestoreMenu	
studentiTableView	
nomeTableColumn	
cognomeTableColumn	
emailTableColumn	
codStudianteTableColumn	
seessoTableColumn	
datadilnascitaTableColumn	
idoneoTableColumn	
operatoriTableView	
emailoperatoreTableColumn	
nomeoperatoreTableColumn	
cognomeoperatoreTableColumn	
codoperatoreTableColumn	
richiestaTableColumn	
listStudenti	
listOperatori	
rowData	
rowDataOperatore	
studenteDAO	
autenticazione	
autenticazioneDAO	
gestore	
gestoreDAO	
areaTematicaDAO	
corso	
corsoDAO	
operatoreDAO	
operatore	
utenteDAO	
loginStage	
Responsibilities:	
Name	Collaborator
Permette al gestore di inviare richieste agli operatori per la gestione dei corsi.	
Permette al gestore di creare, modificare o eliminare dei corsi.	
Permette al gestore di eliminare il proprio account.	
Permette al gestore di visualizzare i dettagli di un corso e le sue statistiche.	

IndirizzoController	
Super Classes: RegistrazioneGestoreController	
Description: Regostrazione indirizzo gestore	
Attributes:	
Name	Description
inserisciButton	
viaTextField	
civicoTextField	
cittaTextField	
provinciaTextField	
siglaTextField	
comuni	
province	
provinciaListner	
Responsibilities:	
Name	Collaborator
Permette la registrazione della sede del gestore e del suo indirizzo	

LoginController	
Description: Rappresenta la Login page	
Attributes:	
Name	Description
loginButton	
emailTextField	
passwordTextField	
loginProgressBar	
gestoriemailTextField	
gestoriPasswordField	
passwordDimenticataLabel	
autenticazioneDAO	
jMetro	
checkEmailListner	
checkEmailGestoriListner	
Responsibilities:	
Name	Collaborator
Gestisce il login si per gli utenti che per i gestore	

RegistrazioneGestoreController	
Super Classes: RegsitrazioneController	
Sub Classes: IndirizzoController	
Description: Registrazioe di un gestore	
Attributes:	
Name	Description
registratigestoreButton	
descrizionegestoreTextArea	
telefonoTextField	
indirizzoTextField	
nomeTextField	
sede	
gestore	
autenticazioneDAO	
gestoreDAO	
sedeDAO	
nomeListner	
descrizioneListner	
Responsibilities:	
Name	Collaborator
Permette la registrazione di un nuovo gestore.	

RegistrazioneController	
Sub Classes: RegistrazioneGestoreController	
Description: Registrazione di un utente	
Attributes:	
Name	Description
emailTextField	
nomeTextField	
cognomeTextField	
dataDatePicker	
comunedinascitaTextField	
seessoChoiceBox	
codicefiscaleTextField	
passwordField	
passwordProgressBar	
repeatpasswordField	
comuni	
utente	
autenticazione	
autenticazioneDAO	
utenteDAO	
generaCodiceListner	
checkEmailListner	
passwordListner	
repeatpasswordListner	
Responsibilities:	
Name	Collaborator
Permette la registrazione di un nuovo utente.	