



PROGETTAZIONE E IMPLEMENTAZIONE DI UN SISTEMA GESTIONALE INTERFACCIATO AD UNA BASE DI DATI

Cristian Carrella
Matricola: N86003677
Alfredo D'Andrea
Matricola: N86003615

Università Federico II, Napoli

INDICE

INDICE	2
OBJECT ORIENTATION	3
Capitolo 1 – PANORAMICA GENERALE.....	3
Descrizione del progetto.....	3
1.1 Analisi del problema.....	3
1.2 Soluzione proposta.....	3
1.3 Organizzazione dei file nella repository	4
Capitolo 2 – FASE DI PROGETTAZIONE.....	5
2.1 Mock-Up.....	5
2.2 UML: Class Diagram.....	6
2.3 Schema concettuale	7
Capitolo 3	8
3.1 Classi.....	8
3.2 Sequence Diagram.....	10
3.3 Class Responsibility Collaboration Card	11

OBJECT ORIENTATION

Capitolo 1 – PANORAMICA GENERALE

Descrizione del progetto

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI Swing, per la gestione di corsi di formazione. Il sistema permette ad un operatore di gestire corsi di formazione. Un corso è caratterizzato da un nome, una descrizione, un tasso di presenze minimo necessario, un numero massimo di partecipanti, e da una o più lezioni. Ciascuna lezione è caratterizzata da un titolo, una descrizione, una durata, e una data e orario di inizio. I corsi possono inoltre essere organizzati in aree tematiche definibili dagli operatori, e un corso può appartenere a più aree tematiche. Gli operatori possono anche iscrivere studenti ai corsi e, per ogni lezione, tenere traccia delle presenze/assenze degli studenti iscritti. Il sistema permette di effettuare interrogazioni avanzate sui corsi erogati, con possibilità di filtraggio per una o più categorie, per data, per parole chiave. Inoltre, il sistema permette di visualizzare, per ciascun corso, statistiche sul tasso di frequenza e un prospetto dove sono evidenziati gli studenti che hanno ottenuto il numero minimo di presenze e sono quindi idonei al superamento del corso.

1.1 Analisi del problema

Il problema consiste nel realizzare un software gestionale. Per poterlo risolvere si è deciso di suddividere il processo realizzativo in diverse fasi:

- Fase di progettazione: realizzazione dell'**Class Diagram** e dei **Mock-Up**.
- Fase decisionale: scelta delle librerie **Swing**, del **Design Patter** da utilizzare, come suddividere il lavoro...
- Realizzazione del database: definizione tabelle, vincoli, trigger...
- Realizzazione interfaccia grafica di base.
- Programmazione delle classi.

L'obiettivo è dare pragmaticità, riutilizzabilità, ordine, funzionalità e propensità all'utilizzo di altri sistemi di conservazione dei dati.

1.2 Soluzione proposta

Per realizzare il progetto abbiamo deciso di attuare le seguenti soluzioni:

1. Utilizzare il Class Diagram per avere una visione chiara di ciò che doveva essere poi realizzato.
2. Utilizzare i Mock-Up per fornire un'idea di come doveva presentarsi il programma alla vista di un utente e come doveva comportarsi alla riduzione/espansione della finestra (renderlo quanto più responsive possibile).
3. Utilizzare le librerie **Swing** poiché oltre ad essere di facile utilizzo sono risultate essere adatte alle interfacce progettate con i Mock-Up
4. Utilizzare 2 Design Pattern:
 - a. **Singleton**
 - b. **DAO**

1.3 Organizzazione dei file nella repository

I file sorgenti (src) sono suddivisi in due cartelle:

- UserGenerator
- Main

UserGenerator è un applicativo dove è possibile registrare un nuovo operatore al quale verrà assegnato un username con il quale potrà accedere al vero e proprio software gestionale.

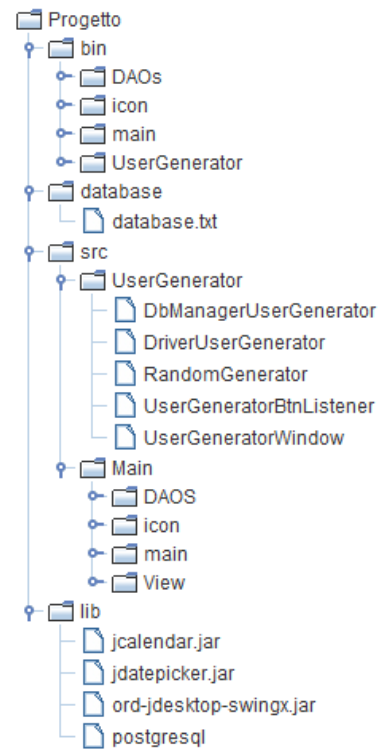
Main è la cartella contenente i file sorgenti in risposta alla traccia assegnata.

La cartella Main è a sua volta suddivisa in:

- DAOS: contenente la realizzazione delle interfacce e le relative implementazioni dei dao.
- icon: contenente tutte le icone che sono state utilizzate all'interno del software.
- main: contenente tutti i file sorgenti relativi alla gestione logico-funzionale del programma.
- view: contenente i file sorgenti di tutti i frame Swing.

Altre cartelle importanti sono:

- Database: nel quale è situato un file di testo (utilizzato dal programma) per poter definire in modo automatico il database (postgres).
- Lib: nel quale sono situate tutte le librerie esterne utilizzate ed aggiunte al ClassPath.

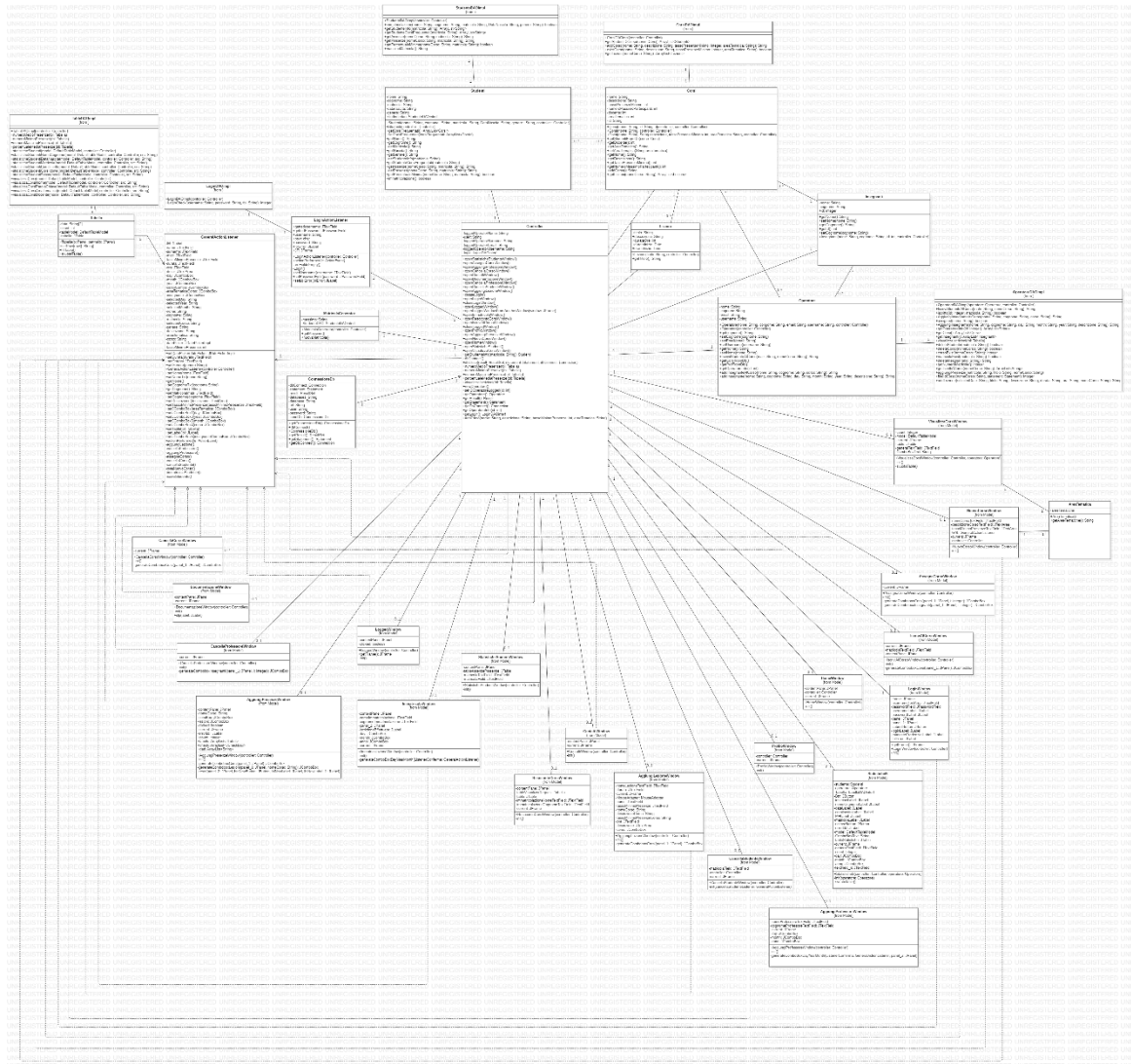


Per l'organizzazione del lavoro di gruppo abbiamo utilizzato una repository su gitHub dalla quale con push e pull gestivamo il versioning del progetto.

2.2 UML: Class Diagram

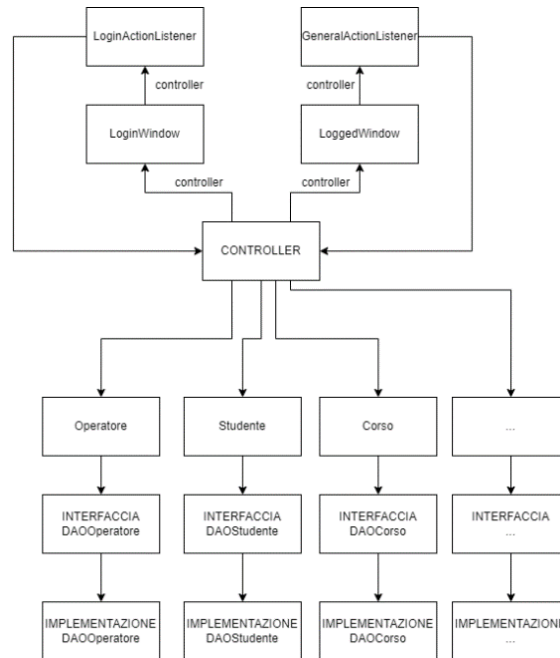
Il secondo passo nella progettazione del software è stato quello di realizzare un class diagram (al di sotto riportato) grazie a cui abbiamo potuto avere un'idea realizzativa su come potessimo organizzare le classi, di quali attributi e metodi avremo avuto bisogno e quindi avere un'idea chiara di ciò che il software doveva compiere.

Ctrl + click sull'immagine per ingrandirla



2.3 Schema concettuale

Nonostante le idee fossero abbastanza chiare grazie ai due passi precedenti abbiamo realizzato uno "schema concettuale" grazie a cui è possibile recepire ancora più rapidamente il metodo funzionale e organizzativo delle classi.



Capitolo 3

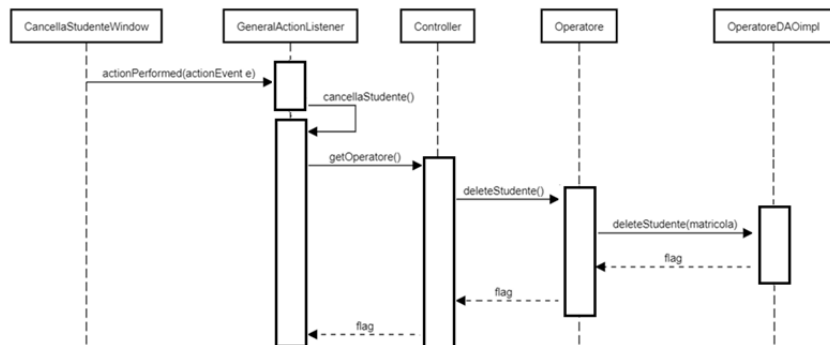
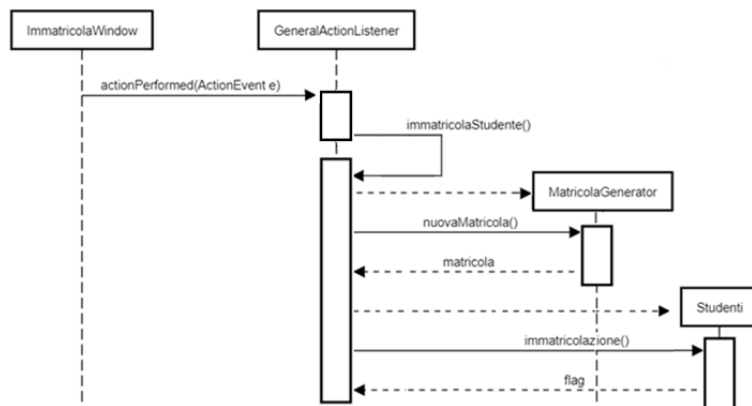
3.1 Classi

Classe	Descrizione
Controller	Classe che permette di gestire in modo efficiente le interazioni tra finestre.
ConnessioneDb	Classe (Singleton) che permette di gestire la connessione al Database
Corsi	Classe che permette di gestire tutte le operazioni riguardanti i corsi.
Operatore	Classe che permette di gestire tutte le operazioni possibili dell'operatore loggato.
Studenti	Classe che permette di gestire tutte le operazioni riguardanti gli studenti.
MatricolaGenerator	Classe che permette di generare una nuova matricola durante l'immatricolazione di un nuovo studente.
Insegnanti	Classe che gestisce tutte le operazioni riguardanti gli insegnanti.
Lezioni	Classe che gestisce tutte le operazioni riguardanti le lezioni.
AreaTematica	Classe contenente tutte le aree tematiche.
LoginActionListener	ActionListener per la finestra di login.
GeneralActionListener	ActionListener che gestisce gli input di tutte le finestre.
Tabella	Classe di gestione della tabella presente nel file "statisticheCorsiWindow".
AggiungiLezioneWindow	Classe che permette di inserire una nuova lezione all'interno di un corso.
AggiungiPresenzeWindow	Classe che permette di aggiungere una nuova presenza di uno studente in una lezione specifica di un corso.
AggiungiProfessoreWindow	Classe che permette di creare un nuovo professore nel database.
AssegnaCorsoWindow	Classe che permette di assegnare un professore ad uno o più corsi.
CancellaCorsoWindow	Classe che permette di eliminare un corso dal sistema.
CancellaprofessoreWindow	Classe che permette la cancellazione di un professore dal database.
CancellaStudenteWindow	Classe che elimina uno studente dall'elenco e lo rimuove da tutti i corsi e lezioni.
ContattiWindow	Classe contenente le informazioni di contatto degli sviluppatori del codice.
DcoumentazioneWindow	Classe che rimanda alla documentazione del software.
HomeWindow	Classe visualizzazione della pagina home.
ImmatricolaWindow	Classe che permette di immatricolare nuovi studenti.
IscriviAlCorsoWindow	Classe che consente di iscrivere uno studente a uno o più corsi.
LoggedWindow	Classe che gestisce tutte le operazioni consentite all'interno del programma.
LoginWindow	Classe che permette l'accesso al sistema.
NuovoCorsoWindow	Classe consente di creare un nuovo corso con tutte le sue specifiche.
ProfiloWindow	Finestra per visualizzare i dati dell'operatore loggato.

<i>ResocontoCorsiWindow</i>	Statistiche degli studenti presenti in un corso per ogni lezione.
<i>StatisticheS</i>	Classe di visualizzazione di tutti gli studenti iscritti con possibilità di ricerca tramite filtraggio di informazioni.
<i>StatisticheStudentiWindow</i>	Classe che permette di visualizzare informazioni sulla carriera di uno studente mediante la ricerca della sua matricola.
<i>VisualizzaCorsiWindow</i>	Finestra di visualizzazione di tutti i corsi con possibilità di ricerca avanzata.
<i>CorsiDAOimpl</i>	Implementazione della interface CorsiDAOinterf per eseguire le query riguardanti i corsi per un database postgres
<i>LoginDAOimpl</i>	Implementazione della classe LoginDAOinterf per eseguire le query riguardanti il login per un database postgres
<i>OperatoreDAOimpl</i>	Implementazione della interface OperatoreDAOinterf per eseguire le query riguardanti le possibili operazioni compiute dall'operatore loggato (cancellazione corso, creazione lezione...)
<i>StudenteDAOimpl</i>	Implementazione della interface StudenteDAOinterf per eseguire le query riguardanti gli studenti per un database postgres
<i>TableDAOimpl</i>	Implementazione della interface TableDAOinterf per eseguire le query riguardanti il riempimento di tutte le tabelle presenti nel software
<i>StatisticheLezioneWindow</i>	Classe di visualizzazione assenze/presenze per ogni lezione di ogni corso

3.2 SEQUENCE DIAGRAM

Sequence diagram di immatricula e cancella studente.



3.3 Class Responsibility Collaboration Card

AggiungiLezioneWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad una lezione Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Corsi GeneralActionListener

AggiungiPresenzeWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad una presenza a lezione Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Lezioni GeneralActionListener Corsi

AggiungiPorfessoreWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad un docente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller GeneralActionListener

AssegnaCorsoWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad un corso e un docente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Insegnanti Corsi GeneralActionListener

CancellaCorsoWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad un corso Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Corsi GeneralActionListener

ProfiloWindow	
<ul style="list-style-type: none"> Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i dati dell'operatore loggato 	<ul style="list-style-type: none"> Controller

ResocontoCorsiWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ai corsi Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller Tabella GeneralActionListener

StatisticheS	
<ul style="list-style-type: none"> Conosce i dati relativi agli studenti Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller Operatore Studenti TableDAOImpl TableDAOInterf GeneralActionListener

StatisticheStudenteWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad uno studente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller Corsi Studenti GeneralActionListener

CancellaProfessoreWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad un docente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Insegnanti GeneralActionListener

CancellaStudenteWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad uno studente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller GeneralActionListener

ContattiWindow	
<ul style="list-style-type: none"> Conosce i dati relativi agli operatori Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller

DocumentazioneWindow	
<ul style="list-style-type: none"> Conosce i dati relativi alla documentazione Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per linkare i suddetti dati 	<ul style="list-style-type: none"> Controller

HomeWindow	
<ul style="list-style-type: none"> Ha lo scopo di interfacciarsi all'utente sotto forma di finestra di pagina home di presentazione 	<ul style="list-style-type: none"> Controller

Studenti	
<ul style="list-style-type: none"> Nome dello studente Cognome dello studente Matricola dello studente Data di nascita dello studente Genere dello studente Permette alle altre classi di ottenere informazioni sullo studente Permette alle altre classi di ottenere tutti gli attributi 	<ul style="list-style-type: none"> StudenteDAOInterf Controller Corsi

Tabella	
<ul style="list-style-type: none"> Permette di creare una nuova tabella riga per riga Permette di svuotare una tabella già esistente 	

VisualizzaCorsiWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ai corsi Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller Operatore AreaTematica TableDAOInterf TableDAOImpl GeneralActionListener

OperatoreDAOimpl	
<ul style="list-style-type: none"> • Permette di iscrivere uno studente ad un corso • Controlla se uno studente è iscritto ad un corso • Assegna un docente all'insegnamento di un corso • Controlla se un corso ha un docente assegnato • Aggiunge un insegnante al database • Permette di visualizzare le informazioni personali dell'operatore loggato • Consente la visualizzazione dei corsi presenti nel database • Consente la visualizzazione dei docenti presenti nel database • Permette di visualizzare gli studenti iscritti ad un determinato corso • Permette di eliminare uno studente dal database • Permette di eliminare un corso dal database • Cerca l'ID di un corso dato un nome in input • Controlla l'esistenza di una matricola presente in un database • Permette di eliminare un docente dal database • Visualizza quanti studenti sono stati immatricolati • Visualizza quanti studenti sono iscritti ad uno specifico corso • Aggiunge una nuova presenza di uno studente a una lezione • Ricava l'ID di un corso • Aggiunge una lezione ad un corso 	<ul style="list-style-type: none"> • Controller • Corsi • Insegnanti • Operatore • Tabella

Interface OperatoreDAOInterf	
• Interfaccia all'utente per accedere ai metodi di OperatoreDAOimpl	

StudenteDAOimpl	
<ul style="list-style-type: none"> • Immatricola un nuovo studente • Mostra le informazioni di uno studente • Visualizza i corsi a cui è iscritto uno studente • Mostra le assenze di uno studente ad un corso • Mostra le presenze di uno studente ad un corso • Controlla se è idoneo a passare un corso date le presenze minime di esso 	<ul style="list-style-type: none"> • Controller

Corsi	
<ul style="list-style-type: none"> • ID del corso • Nome del corso • Descrizione del corso • Statistiche di studenti nel corso • Area tematica del corso • Insegnante del corso • Lezioni del corso • Permette di settare una specifica area tematica da associare ad un corso • Permette di ottenere ogni singolo attributo della classe da chiunque la richiami • Permette di aggiungere un corso 	<ul style="list-style-type: none"> • Insegnanti • CorsiDAOInterf • Controller • Lezioni • Studenti

GeneralActionListener	
• ActionListener che gestisce gli input di tutte le finestre.	

Insegnanti	
<ul style="list-style-type: none"> • ID del docente • Nome del docente • Cognome del docente • Permette di settare i suoi attributi • Permette ad altre classi di ottenere i suoi attributi mediante chiamata 	<ul style="list-style-type: none"> • Controller

Lezioni	
<ul style="list-style-type: none"> • Titolo della lezione • Descrizione della lezione • Durata della lezione • Orario di inizio lezione • Data dello svolgimento della lezione • Corso contenente la lezione • Permette di far ottenere il titolo della lezione dalle classi chiamanti 	<ul style="list-style-type: none"> • Corsi • Controller

LoginActionListener	
<ul style="list-style-type: none"> • Nome utente dell'operatore loggato • Password dell'operatore loggato • Permette l'accesso al sistema da parte di un operatore 	<ul style="list-style-type: none"> • Controller

MatricolaGenerator	
• Classe che permette di generare una nuova matricola durante l'immatricolazione di un nuovo studente.	• StudenteDAOInterf

Operatore	
<ul style="list-style-type: none"> • Nome dell'operatore • Cognome dell'operatore • Email dell'operatore • Username dell'operatore • Permette di settare tutti i suoi attributi • Permette di ottenere i suoi attributi dalle classi chiamanti 	<ul style="list-style-type: none"> • OperatoreDAOInterf • Controller • Corsi • Insegnanti

Controller	
<ul style="list-style-type: none"> • Può aprire qualsiasi finestra • Può chiudere qualsiasi finestra aperta 	<ul style="list-style-type: none"> • Operatore • LoginDAOInterf • TableDAOInterf • ProfiloWindow • LoginWindow • LoginWindow • ImmatricolaWindow • ResocontoCorsiWindow • IscrivitiAlCorsoWindow • NuovoCorsoWindow • AssegnaCorsoWindow • AggiungiProfessoreWindow • CancellaCorsoWindow • CancellaProfessoreWindow • CancellaStudenteWindow • AggiungiLezioneWindow • StatisticheStudenteWindow • HomeWindow • ContattiWindow • DocumentazioneWindow • StatisticheS • AggiungiPresenzeWindow • VisualizzaCorsiWindow • ConnessioneDb • Studenti • Corsi

AreaTematica	
• Classe contenente tutte le aree tematiche (per assunzione) già definite.	

ConnessioneDb	
• Consente la connessione al database.	

ImmatricolaWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad uno studente Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller GeneralActionListener

IscriviAlCorso	
<ul style="list-style-type: none"> Conosce i dati relativi ad uno studente e un corso Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller Corsi GeneralActionListener

LoggedWindow	
<ul style="list-style-type: none"> Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per permettere all'utente la navigazione nel sistema 	<ul style="list-style-type: none"> Controller GeneralActionListener

LoginWindow	
<ul style="list-style-type: none"> Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i dati di accesso per entrare nel sistema 	<ul style="list-style-type: none"> Controller LoginActionListener

NuovoCorsoWindow	
<ul style="list-style-type: none"> Conosce i dati relativi ad un corso Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per chiedere i suddetti dati 	<ul style="list-style-type: none"> Controller AreaTematica GeneralActionListener

StatisticheLezioniWindow	
<ul style="list-style-type: none"> Conosce i dati relativi alle statistiche sulle lezioni Ha lo scopo di interfacciarsi all'utente sotto forma di finestra per mostrare i suddetti dati 	<ul style="list-style-type: none"> Controller TableDAOImpl TableDAOInterf

AreaTematica	
<ul style="list-style-type: none"> Classe che si occupa di gestire la classificazione dei corsi in una o più aree tematiche 	<ul style="list-style-type: none"> Controller

Interface StudenteDAOInterf	
<ul style="list-style-type: none"> Interfaccia all'utente per accedere ai metodi di StudenteDAOImpl 	

TableDAOImpl	
<ul style="list-style-type: none"> Crea tabelle variabili a seconda dei valori in input dei vari metodi presenti nella classe 	<ul style="list-style-type: none"> Controller Tabella

Interface TableDAOInterf	
<ul style="list-style-type: none"> Interfaccia all'utente per accedere ai metodi di TableDAOImpl 	

CorsiDAOImpl	
<ul style="list-style-type: none"> Permette di visualizzare gli studenti di un corso Permette di aggiungere un nuovo corso Controlla se esiste un corso Visualizza le lezioni 	<ul style="list-style-type: none"> Controller Studenti CorsiDAOInterf Lezioni Corsi

Interface CorsiDAOInterf	
<ul style="list-style-type: none"> Interfaccia all'utente per accedere ai metodi di CorsiDAOImpl 	

LoginDAOImpl	
<ul style="list-style-type: none"> Si occupa di verificare le credenziali nel database dell'operatore che cerca di accedere al sistema 	<ul style="list-style-type: none"> LoginDAOInterf LoginActionListener

Interface LoginDAOInterf	
<ul style="list-style-type: none"> Interfaccia all'utente per accedere ai metodi di LoginDAOImpl 	

NOTA:

NEL POPOLAMENTO DEL DATABASE SONO STATI INSERITI SOLO CORSI UNIVERSITARI A SCOPO DIMOSTRATIVO MA IL SOFTWARE FUNZIONA PER TUTTI I TIPI DI CORSI FORMATIVI.