



Università degli Studi di Salerno

Corso di Ingegneria del Software

Coffee Object Design Document Versione 1.3



Partecipanti al progetto

Nome	Matricola
Vincenzo Alessandro Cavaso	05102/01132
Demia Massaro	05103/00190
Antonio Sanfelice	05102/00997

Scritto da:	Demia Massaro
Revisionato da:	Antonio Sanfelice
Approvato da:	Vincenzo Alessandro Cavaso

Revision History

Data	Versione	Descrizione	Autore
8/10/2008	1.1	Introduzione del documento	Team
15/10/2008	1.2	Packages	Team
22/10/2008	1.3	Class Interfaces	Team

Indice

1. Introduzione.....	4
1.1 Compromessi dell'Object Design.....	4
1.1.1 Comprensibilità vs tempo.....	4
1.1.2 Sicurezza vs efficienza.....	4
1.2 Linee guida per la documentazione delle interfacce.....	4
1.3 Definizioni, acronimi ed abbreviazioni.....	5
1.4 Riferimenti.....	5
2. Packages.....	6
2.1 Package Coffee.....	6
2.2 Package Studente.....	6
2.3 Package Docente.....	7
2.3 Package Operatore.....	7
3. Class Interfaces.....	8
3.1 Studente.....	8
3.1.1 Gestione Analisi Carriera.....	8
3.1.2 Gestione Propri Esami.....	10
3.2 Docente.....	12
3.2.1 Gestione Propri Appelli.....	12
3.3 Operatore.....	15
3.3.1 Gestione Esami.....	15
3.3.2 Gestione Appelli.....	18
3.3.3 Gestione Utenti.....	21

1. Introduzione

1.1 Compromessi dell'Object Design

Per ridurre i tempi di rilascio è stato deciso di utilizzare alcune componenti software già pronte per gestire alcune delle funzionalità più complesse del sistema. In particolare verrà usata la componente software open-source JDBC per l'interfacciamento col database.

1.1.1 Comprensibilità vs tempo

Il codice verrà arricchito da commenti che aiutino la comprensione del sorgente. Tali commenti però, verranno aggiunti solo a quelle parti del codice ambigue o di difficile comprensione, in modo da non aumentare troppo i tempi di sviluppo

1.1.2 Sicurezza vs efficienza

Essendo il sistema pensato per funzionare in una rete locale, non è stato dato molto peso al problema della sicurezza: si ritiene sufficiente l'uso di un meccanismo di autenticazione Login/Password. Il mancato uso di meccanismi di crittografia delle comunicazioni migliorerà, anche se minimamente, le prestazioni del sistema.

1.2 Linee guida per la documentazione delle interfacce

Vengono qui elencate le linee guida che gli sviluppatori dovranno seguire durante la stesura del codice sorgente:

Stile di programmazione

- Lo spazio di indentazione delle classi deve essere di un tab

Naming Convention

- I nomi delle classi devono cominciare con una lettera maiuscola, e che anche le parole seguenti all'interno del nome devono cominciare con una lettera maiuscola. I nomi delle classi inoltre devono fornire informazioni sul loro scopo.
Es. UtenteRegistrato
- I nomi dei metodi devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola.
Es. aggiungiEsame()
- I nomi dei metodi per l'accesso e la modifica delle variabili dovranno essere del tipo *getNomeVariabile()*, *setNomeVariabile()* o nel caso di valori booleani *isNomeVariabileBooleana()*

- I nomi delle variabili devono cominciare con una lettera minuscola, e le parole seguenti con la lettera maiuscola. *Es. annodiscrizione*
- I nomi delle costanti devono essere scritti a lettere maiuscole, e le parole separate da un underscore. *Es. static final int ESEMPIO_COSTANTE = k*

Documentazione

- Ogni classe deve avere una breve spiegazione del suo scopo, dopo le istruzioni di import. Devono essere inoltre indicati l'autore della classe e altre informazioni utili utilizzando gli appropriati tag Javadoc

```
/** descrizione della classe  
 *  
 * @author nome dell'autore  
 * ...  
 */
```

- La descrizione del metodo deve apparire prima di ogni dichiarazione di metodo e deve descriverne lo scopo. Devono essere elencati i parametri del metodo, i valori di ritorno ed eventualmente le eccezioni che possono essere lanciate, utilizzando gli appropriati tag di Javadoc

```
/**  
 * Descrizione dello scopo del metodo  
 * @param Param1 – descrizione parametro 1  
 * @return descrizione del valore di ritorno  
 */
```

- Le variabili devono essere documentate solo se il loro scopo non è immediatamente intuibile. Possono essere usati sia commenti Javadoc che normali commenti

1.3 Definizioni, acronimi ed abbreviazioni

SDD : System Design Document

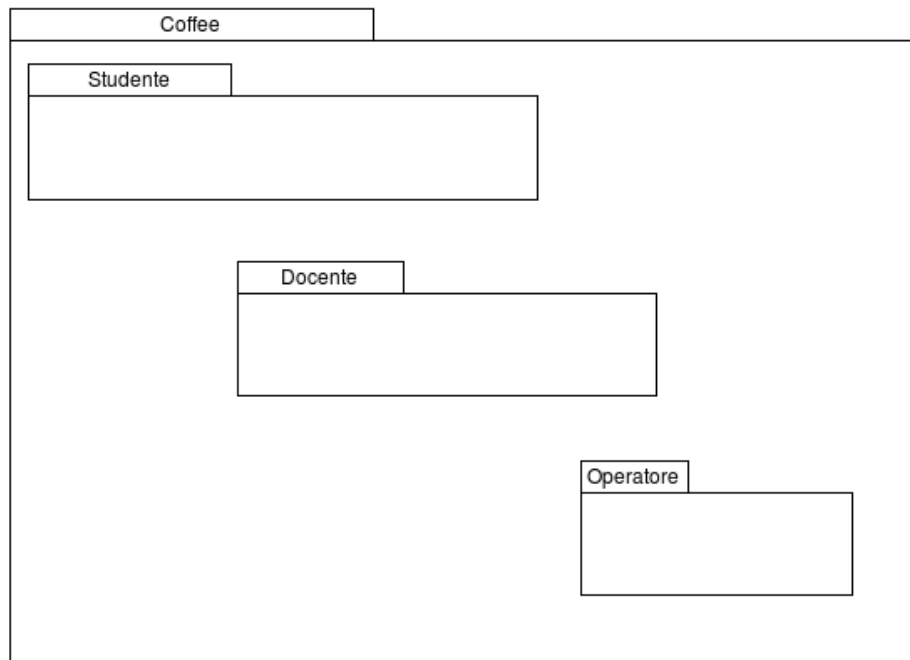
JDBC: Java DataBase Connectivity

RMI: Remote Method Invocation

1.4 Riferimenti

Coffee SDD

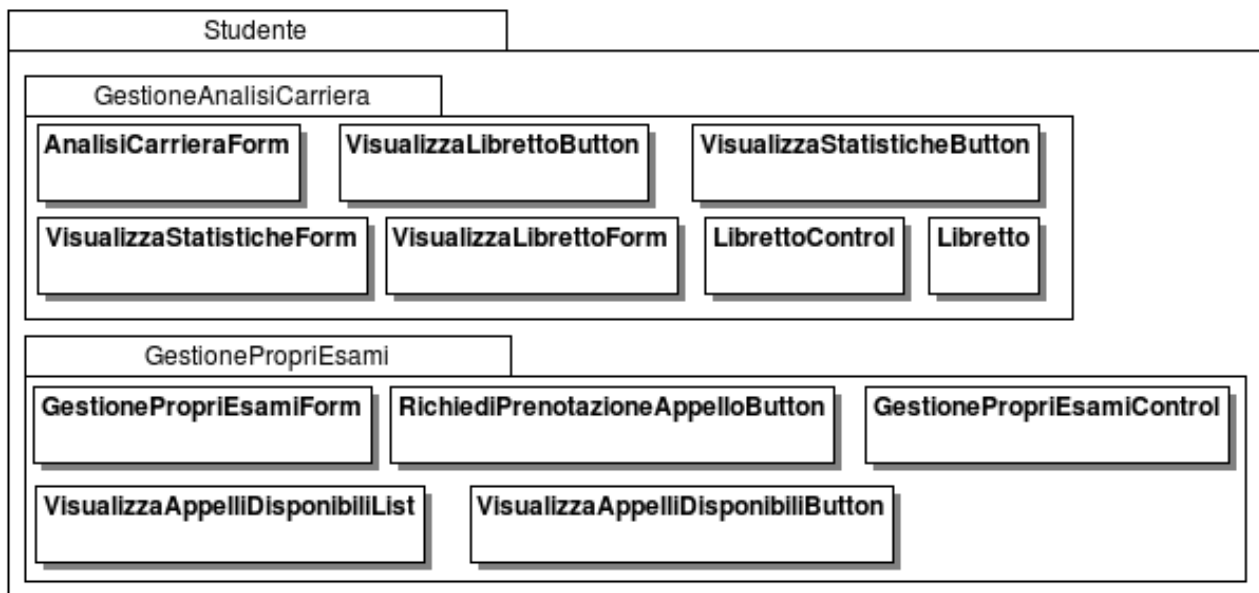
2. Packages



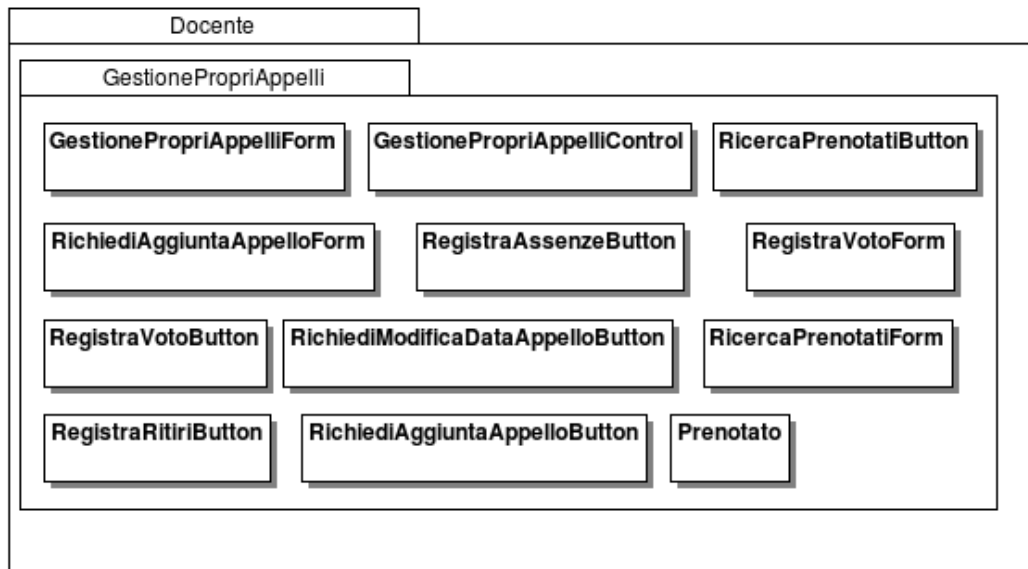
2.1 Package Coffee

Il package Coffee è composto da tre sotto-package che rispecchiano la divisione in sottosistemi individuata durante il system design. Verranno ora analizzati i singoli sotto-package.

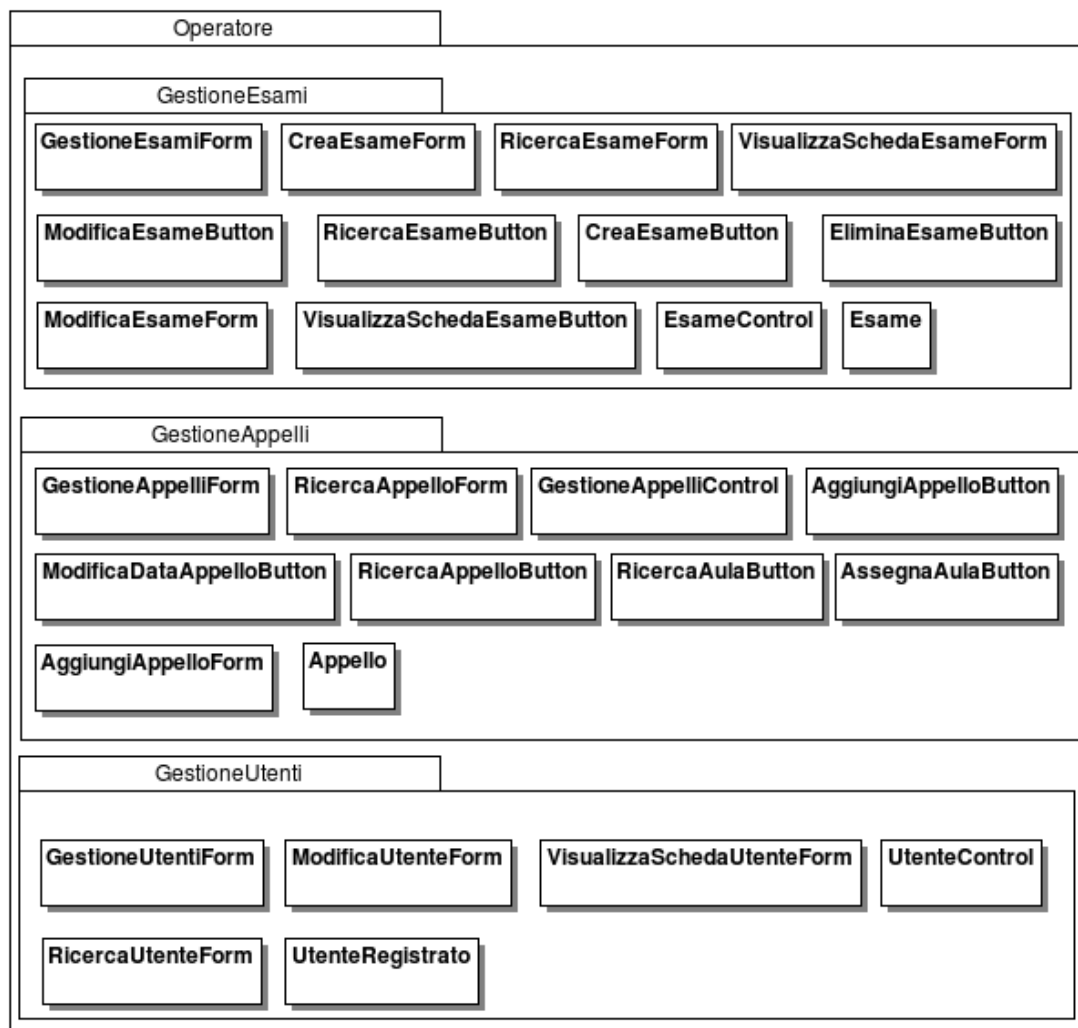
2.2 Package Studente



2.3 Package Docente



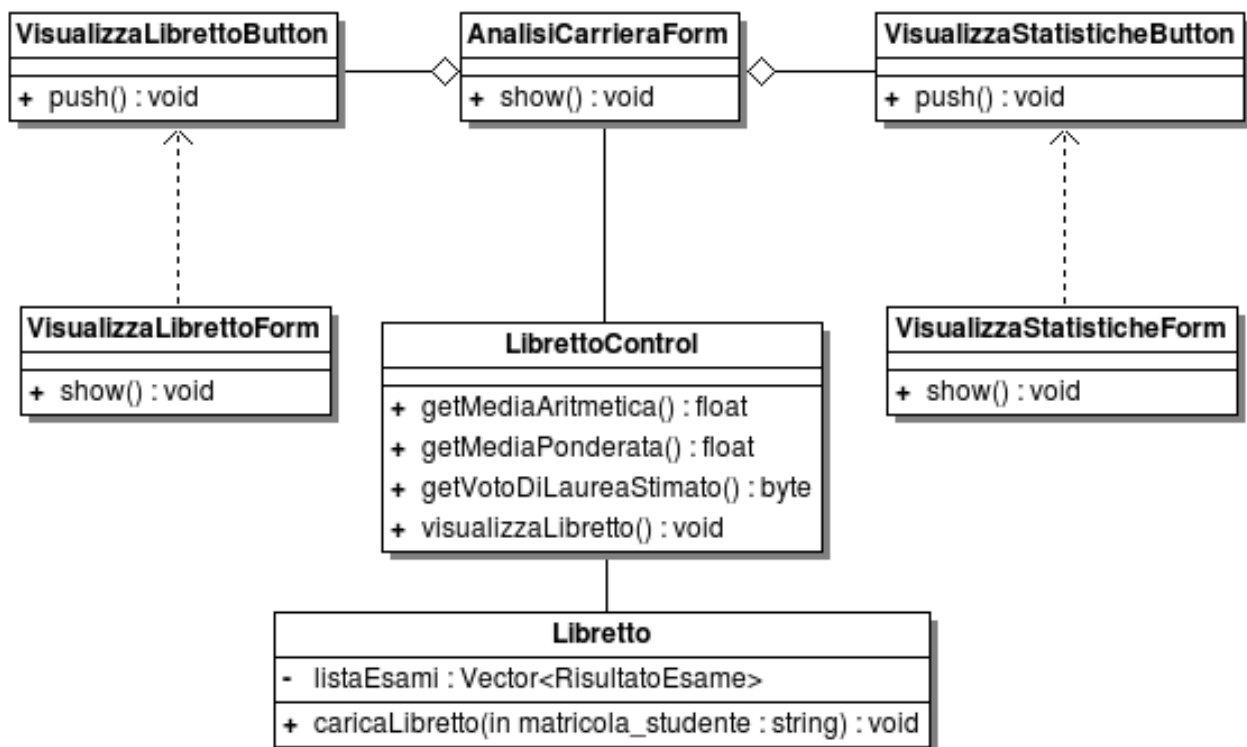
2.3 Package Operatore



3. Class Interfaces

3.1 Studente

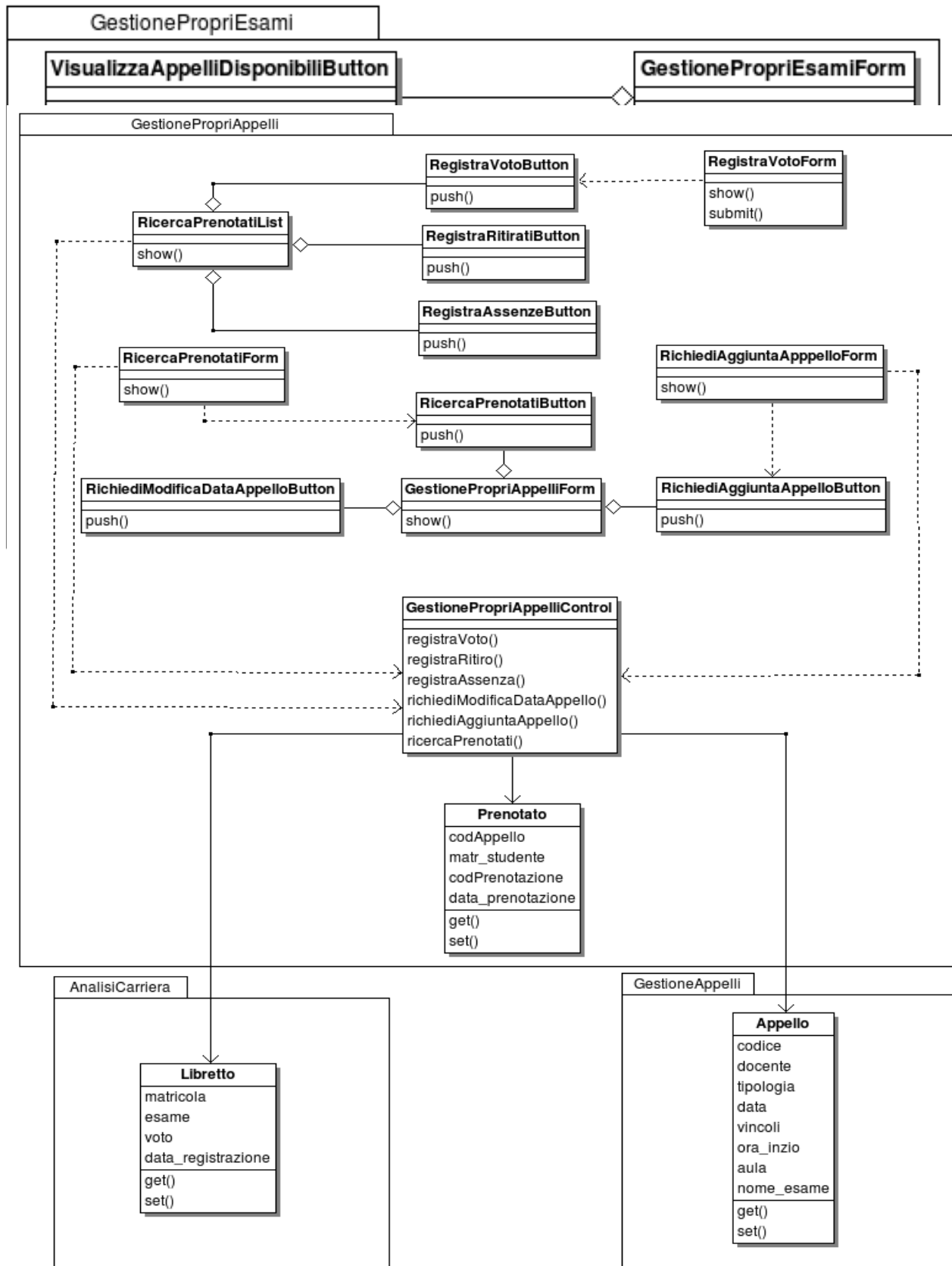
3.1.1 Gestione Analisi Carriera



Class	<i>Libretto</i>	Autore: <i>Demia Massaro</i>	
Descrizione	La classe che modella il libretto di uno studente, è' una vista dell'entità libretto presente nel database.		
Dipendenza	Nessuna		
Attributi			
Nome	Accesso	Descrizione	
listaEsame	privato	La lista degli esami presenti sul libretto	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
loadDataFromData Base	Pubblico	Carica i dati del libretto di un dato studente dal database	SQLException

Class	<i>LibrettoControl</i>	Autore <i>Demia Massaro</i>	
Descrizione	Classe che modella l'oggetto control che si occupa di recuperare dati dall'entity Libretto.		
Dipendenza	Libretto		
Attributi			
Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
getMediaAritmetica	Pubblico	Metodo che calcola la media aritmetica di un dato studente	N/A
getMediaPonderata	Pubblico	Metodo che calcola la media ponderata di un dato studente	N/A
getVotoDiLaureaStimato	Pubblico	Metodo che calcola il voto di laurea stimato di un dato studente	N/A
visualizzaLibretto	Pubblico	Metodo che visualizza il libretto di un dato studente	N/A

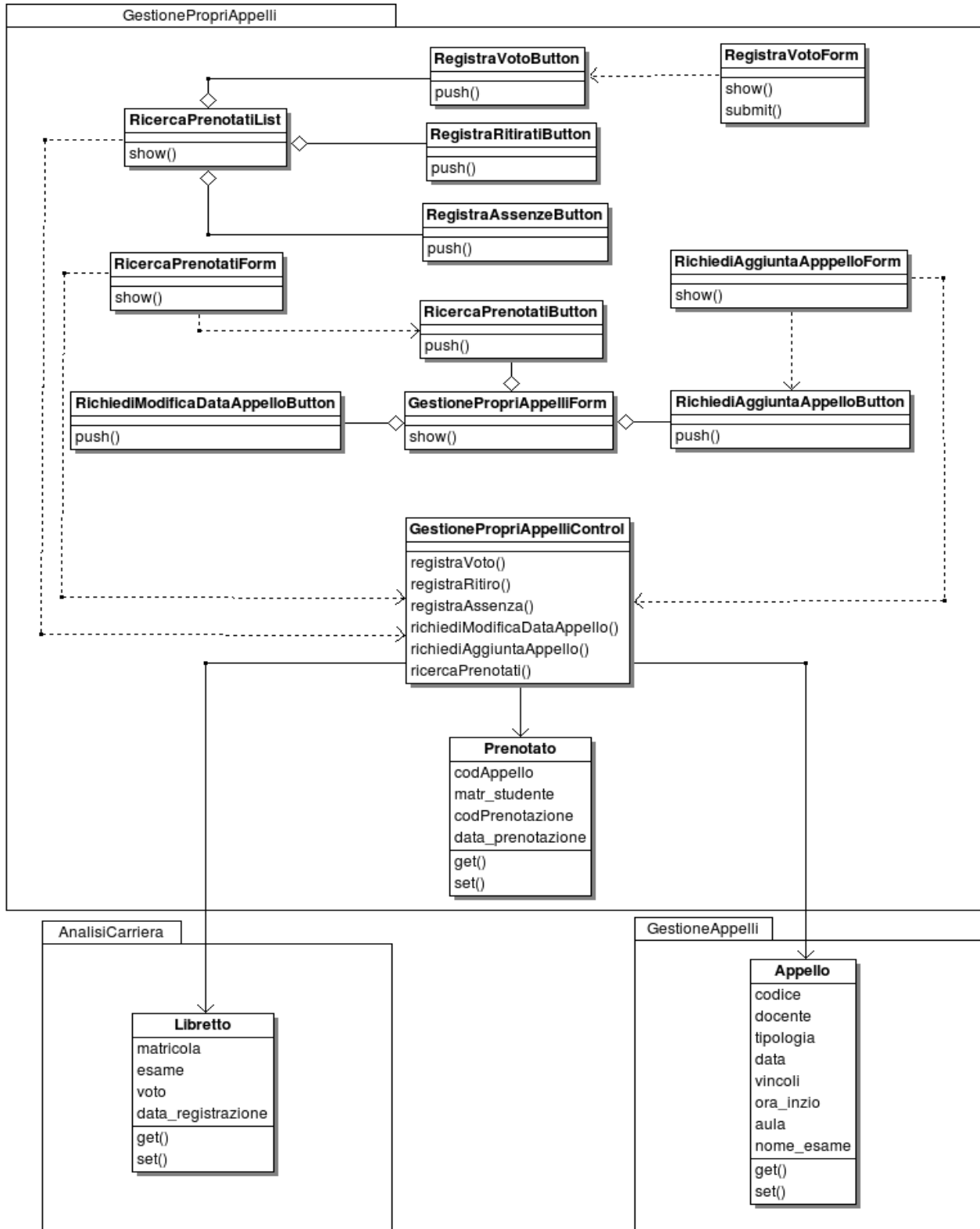
3.1.2 Gestione Propri Esami



Class	<i>GestionePropriEsamiControl</i>	Autore Demia Massaro	
Descrizione	Classe che modella l'oggetto control che si occupa di interagire con gli appelli degli esami di un dato studente		
Dipendenza	Esame, Appello		
Attributi			
Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
richiediPrenotazioneAppello	Pubblico	Metodo che permette ad uno studente di richiedere la prenotazione ad un esame	N/A
visualizzaAppelliDisponibili	Pubblico	Metodo che visualizza gli appelli disponibile per un dato studente	N/A
prenotaEsame	Pubblico	Metodo che fa prenotare uno studente ad un esame	SQLException

3.2 Docente

3.2.1 Gestione Propri Appelli



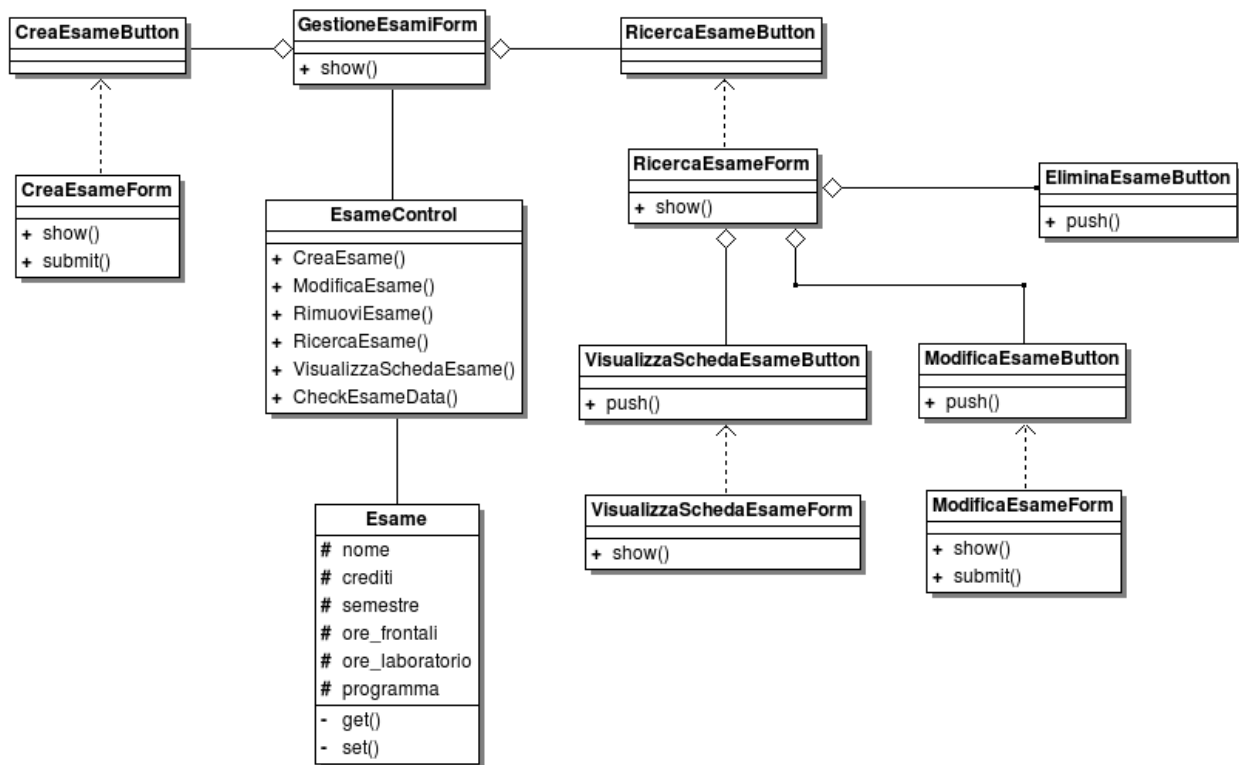
Class	<i>Prenotato</i>	Autore Antonio Sanfelice	
Descrizione	Classe che modella l'entità prenotato		
Dipendenza	Nessuna.		
Attributi			
Nome	Accesso	Descrizione	
nome	Privato	Attributo che descrive il nome dello studente prenotato	
cognome	Privato	Attributo che descrive il cognome dello studente prenotato	
matrStudente	Privato	Attributo che descrive la matricola dello studente prenotato	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
SetNome	Pubblico	Imposta il nome dello studente prenotato	N/A
getNome	Pubblico	Restituisce il nome dello studente prenotato	N/A
setCognome	Pubblico	Imposta il cognome dello studente prenotato	N/A
getCognome	Pubblico	Restituisce il cognome dello studente prenotato	N/A
setMatricola	Pubblico	Imposta la matricola dello studente prenotato	N/A
getMatricola	Pubblico	Restituisce la matricola dello studente prenotato	N/A

Class	<i>GestionePropriAppelliControl</i>	Autore Antonio Sanfelice
Descrizione	Classe che modella l'oggetto Control che interagisce con l'entity appello	
Dipendenza	Appello	
Attributi		

Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
creaAppello	Pubblico	Metodo che permette la creazione di un appello per un dato esame	N/A
ricercaPrenotati	Pubblico	Metodo che permette di ricercare i prenotati ad un dato appello	SQLException

3.3 Operatore

3.3.1 Gestione Esami

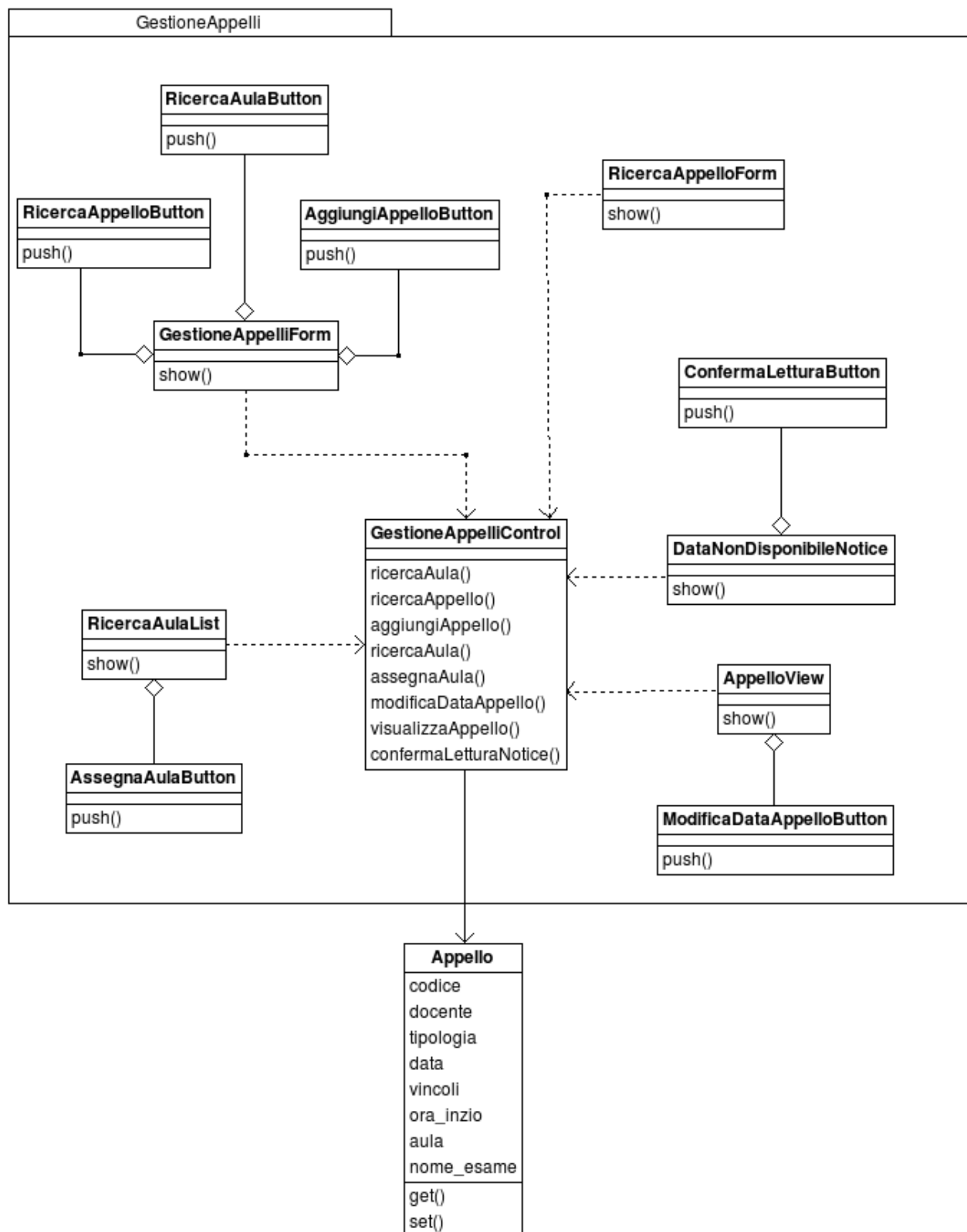


Class	<i>Esame</i>	Autore <i>Vincenzo Alessandro Cavaso</i>	
Descrizione	Classe che modella l'entità esame		
Dipendenza	Nessuna		
Attributi			
Nome	Accesso	Descrizione	
nome	Privato	Rappresenta il nome dell'esame	
semestre	Privato	Rappresenta il semestre dell'esame	
crediti	Privato	Rappresenta i crediti dell'esame	
laboratorio	Privato	Rappresenta la presenza di un laboratorio associato all'esame	
contenuti	Privato	Rappresenta la descrizione dei contenuti didattici di un esame	
obiettivi	Privato	Rappresenta la descrizione degli obiettivi di un esame	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
getNome	Pubblico	Restituisce il nome dell'esame	N/A
setNome	Pubblico	Imposta il nome dell'esame	N/A
getSemestre	Pubblico	Restituisce il semestre dell'esame	N/A
setSemestre	Pubblico	Imposta il semestre dell'esame	N/A
getCrediti	Pubblico	Restituisce il numero di crediti dell'esame	N/A
setCrediti	Pubblico	Imposta il numero di crediti dell'esame	N/A
getLaboratorio	Pubblico	Restituisce la presenza di un laboratorio associato ad un esame	N/A
setLaboratorio	Pubblico	Imposta l'associazione ad un laboratorio di un esame	N/A
getContenuti	Pubblico	Restituisce i contenuti dell'esame	N/A

setContenuti	Pubblico	Imposta i contenuti dell'esame esame	N/A
getObiettivi	Pubblico	Restituisce gli obiettivi dell'esame	N/A
setObiettivi	Pubblico	Imposta gli obiettivi dell'esame	N/A

Class	<i>EsameControl</i>	Autore <i>Vincenzo Alessandro Cavaso</i>	
Descrizione	Classe che modella l'oggetto control che interagisce con l'entity esame		
Dipendenza	Esame		
Attributi			
Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
creaEsame	Private	Metodo che crea un esame	N/A
ModificaEsame	Private	Metodo che modifica un esame esistente	N/A
eliminaEsame	Private	Metodo che elimina un esame esistente	N/A
ricercaEsame	Private	Metodo che permette di effettuare una ricerca sugli esami esistenti	N/A

3.3.2 Gestione Appelli

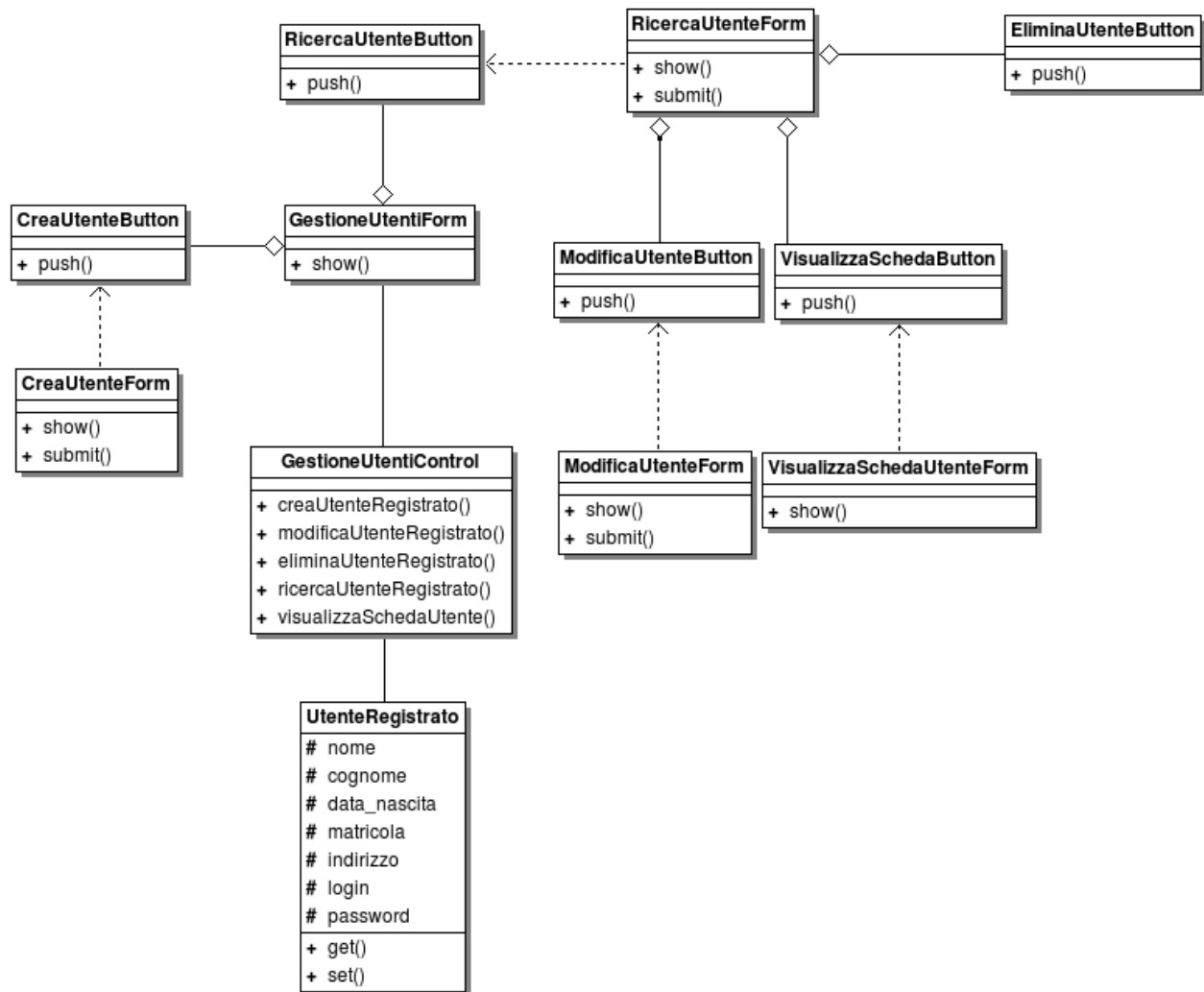


Class	<i>Appello</i>	Autore Antonio Sanfelice	
Descrizione	Classe che modella l'entità appello		
Dipendenza	Nessuna		
Attributi			
Nome	Accesso	Descrizione	
esame	Privato	Rappresenta l'entità esame	
data	Privato	Rappresenta la data dell'esame	
docente	Privato	Rappresenta il docente che detiene l'esame	
tipologia	Privato	Rappresenta la tipologia di esame	
oraInizio	Privato	Rappresenta l'ora di inizio dell'esame	
durata	Privato	Rappresenta la durata dell'esame	
vincoli	Privato	Rappresenta i vincoli di un esame	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
getEsame	Pubblico	Restituisce l'esame associato all'appello	N/A
setEsame	Pubblico	Imposta l'esame associato all'appello	N/A
getData	Pubblico	Restituisce la data dell'appello	N/A
setData	Pubblico	Imposta la data dell'appello	N/A
getDocente	Pubblico	Restituisce il docente associato all'appello	N/A
setDocente	Pubblico	Imposta il docente all'appello	N/A
getTipologia	Pubblico	Restituisce la tipologia dell'appello	N/A
setTipologia	Pubblico	Imposta la tipologia dell'appello	N/A
getOraInizio	Pubblico	Restituisce l'orario di inizio dell'appello	N/A
setOraInizio	Pubblico	Imposta l'orario di inizio dell'appello	N/A
getDurata	Pubblico	Restituisce la durata dell'appello	N/A

setDurata	Pubblico	Imposta la durata dell'appello	N/A
getVincoli	Pubblico	Restituisce i vincoli dell'appello	N/A
setVincoli	Pubblico	Imposta i vincoli dell'appello	N/A

Class	<i>AppelloControl</i>	Autore Antonio Sanfelice	
Descrizione	Classe che modella l'oggetta control che interagisce con l'entity Appello		
Dipendenza	Appello		
Attributi			
Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
aggiungiAppello	Pubblico	Metodo che trasforma una richiesta di aggiunta appello in un appello	N/A
modificaAppello	Pubblico	Metodo che modifica un appello esistente	SQLException
ricercaAppello	Pubblico	Metodo che permette di effettuare una ricerca tra gli appelli esistenti	SQLException
ricercaPropriAppelli	Pubblico	Metodo che permette ad un docente di effettuare una ricerca tra i proprio appelli	SQLException
riceviRichiestaAggiuntaAppello	Pubblico	Metedo che permette di leggere una richiesta di aggiunta appello	FileNotFoundException

3.3.3 Gestione Utenti



Class	<i>UtenteRegistrato</i>	Autore <i>Cavaso Vincenzo Alessandro</i>	
Descrizione	Classe che modella l'utente generico del sistema		
Dipendenza	Nessuna		
Attributi			
Nome	Accesso	Descrizione	
nome	Privato	Rappresenta il nome dell'utente	
cognome	Privato	Rappresenta il cognome dell'utente	
dataNascita	Privato	Rappresenta la data di nascita dell'utente	
indirizzo	Privato	Rappresenta l'indirizzo dell'utente	
matricola	Privato	Rappresenta la matricola dell'utente	
login	Privato	Rappresenta la login dell'utente	
password	Privato	Rappresenta la password dell'utente	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
getNome	Pubblico	Restituisce il nome dell'utente	N/A
setNome	Pubblico	Imposta il nome dell'utente	N/A
getCognome	Pubblico	Restituisce il cognome dell'utente	N/A
setCognome	Pubblico	Imposta il cognome dell'utente	N/A
getDataNascita	Pubblico	Restituisce la data di nascita dell'utente	N/A
setDataNascita	Pubblico	Imposta la data di nascita dell'utente	N/A
getIndirizzo	Pubblico	Restituisce l'indirizzo dell'utente	N/A
setIndirizzo	Pubblico	Imposta l'indirizzo dell'utente	N/A
getMatricola	Pubblico	Restituisce la matricola dell'utente	N/A
setMatricola	Pubblico	Imposta la matricola dell'utente	N/A
getLogin	Pubblico	Restituisce la login dell'utente	N/A
setLogin	Pubblico	Imposta la login dell'utente	N/A
getPassword	Pubblico	Restituisce la password	N/A

		dell'utente	
setPassword	Pubblico	Imposta la password dell'utente	N/A

Class	<i>Studente</i>	Autore <i>Demia Massaro</i>	
Descrizione	Modella l'entità Studente		
Dipendenza	<i>UtenteRegistrato</i>		
Attributi			
Nome	Accesso	Descrizione	
annoImmatricolazione	Privato	Rappresenta l'anno di immatricolazione dello studente	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
setAnnoImmatricolazione	Pubblico	Imposta l'anno di immatricolazione dello studente	N/A
getAnnoImmatricolazione	Pubblico	Restituisce l'anno di immatricolazione dello studente	N/A

Class	<i>Docente</i>	Autore <i>Antonio Sanfelice</i>	
Descrizione	Classe che modella l'entità Docente		
Dipendenza	<i>UtenteRegistrato</i>		
Attributi			
Nome	Accesso	Descrizione	
Tipo	Privato	Rappresenta il tipo di contratto del docente	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
getTipo	Pubblico	Restituisce il tipo di contratto del docente	N/A
setTipo	Pubblico	Imposta il tipo di contratto del docente	N/A

Class	<i>Operatore</i>	Autore <i>Vincenzo Alessandro Cavaso</i>	
--------------	------------------	---	--

Descrizione	Modella l'entità operatore		
Dipendenza	<i>UtenteRegistrato</i>		
Attributi			
Nome	Accesso	Descrizione	
AnnoDiImmatricolazione	Pubblico	Rappresenta l'anno in cui è stato immatricolato l'operatore	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
get	Pubblico	Restituisce l'anno di immatricolazione	N/A
set	Pubblico	Imposta l'anno di immatricolazione	N/A

Class	<i>GestioneUtentiControl</i>	Autore <i>Vincenzo Alessandro Cavaso</i>	
Descrizione	Classe che modella il control per la gestione degli utenti.		
Dipendenza	Nessuna		
Attributi			
Nome	Accesso	Descrizione	
N/A	N/A	N/A	
Metodi			
Nome	Accesso	Descrizione	Eccezioni
riceviRichiestaAggiuntaAccount	Privato	Metodo che permette di ricevere la richiesta di aggiunta di un nuovo account	N/A
visualizzaCodaRichieste	Privato	Metodo che permette di visualizzare tutte le richieste di un nuovo account	N/A
ricercaUtenteRegistrato	Privato	Metodo che permette di effettuare una ricerca sugli utenti registrati al sistema	N/A
creaUtenteRegistrato	Privato	Metodo che crea un nuovo utente registrato nel sistema	N/A
eliminaUtenteRegis	Privato	Metodo che elimina un utente	N/A

trato		registrato dal sistema	
modificaUtenteRegi strato	Privato	Metodo che crea modifica utente registrato nel sistema	N/A