



# Università degli Studi di Salerno

Corso di Ingegneria del Software

## Coffee System Design Document Versione 2.0



**Partecipanti al progetto**

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

<b>Scritto da:</b>	Vincenzo Alessandro Cavaso
<b>Revisionato da:</b>	Demia Massaro
<b>Approvato da:</b>	Antonio Sanfelice

**Revision History**

<b>Data</b>	<b>Versione</b>	<b>Descrizione</b>	<b>Autore</b>
1/07/2008	1.0	Introduzione del sistema	Team
9/07/2008	1.1	Architettura del sistema proposto	Team
16/07/2008	1.2	Gestione del controllo globale	Team
22/07/2008	1.3	Boundary Conditions	Team
3/09/2008	2.0	Servizi dei sottosistemi	Team

## Indice

1. Introduzione.....	4
1.1 Scopo del sistema.....	4
1.2 Obiettivi Desing.....	4
1.2.1 Performance Criteria.....	4
1.2.2 Dependability Criteria.....	5
1.2.3 Maintenance Criteria.....	6
1.3 Definizioni, Acronimi e Abbreviazioni.....	7
1.4 Riferimenti.....	7
2. Architettura del sistema proposto.....	8
2.1 Subsystem decomposition.....	8
2.2 Hardware/Software mapping.....	13
2.3 Gestione dei Dati Persistenti.....	14
2.4 Controllo degli accessi e sicurezza.....	19
2.4.1 Controllo degli accessi.....	19
2.4.2 Sicurezza.....	22
2.5 Gestione del controllo globale.....	23
2.6 Boundary conditions .....	24
2.6.1 Configuration.....	24
2.6.2 Start up – Shut down use cases.....	25
2.6.2.1 Start Up.....	25
2.6.2.2 Shut down.....	27
2.6.2.3 RegistraOperatore:.....	28
2.6.3 Exception handling.....	28
3. Servizi dei Sottosistemi.....	29

# 1. Introduzione

## 1.1 Scopo del sistema

Lo scopo del progetto è la realizzazione di un sistema software gestionale e di supporto alla carriera universitaria degli studenti. Il sistema software COFFEE presenta in particolare vincoli di robustezza, portabilità, facilità d'uso ed altri che saranno descritti in seguito.

## 1.2 Obiettivi Desing

Il sistema proposto cerca di rispettare la maggior parte degli obiettivi di design proponendo dei buoni compromessi (trade-off) fra quelli che sono in contrasto fra loro.

### 1.2.1 Performance Criteria

Criteri	Definizioni
Tempo di risposta	<p>I tempi di risposta sono mediamente di 10 secondi. Queste prestazioni sono garantite considerando un carico di lavoro non molto elevato.</p> <p>Le prestazioni del sistema sono ottime per le operazioni più frequenti.</p>
Memoria	<p>Il sistema richiede elevati quantitativi di memoria a lungo termine, dovuti alla necessità di archiviare dati persistenti negli anni. Per un'esecuzione ottimale del sistema, sono richiesti, inoltre, almeno 256 Megabyte di memoria ram.</p>

## 1.2.2 Dependability Criteria

Criteri	Definizioni
Robustezza	Quasi la totalità degli input prevede controlli che ne verifichino la correttezza a run-time.
Affidabilità	In casi di interruzione al server i client dovranno attendere il ripristino della connessione al server per effettuare successive operazioni. Il sistema garantisce l'atomicità delle operazioni.
Tolleranza ai crash	Il sistema dovrà essere in grado di rispondere in maniera adeguata ad eventuali comportamenti errati dell'utente che interagisce con esso, in quanto la principale forma di interazione consiste nell'immissione di dati in determinati form, aspetto da tenere in considerazione per impedire l'inserimento di informazioni errate o inconsistenti che potrebbero portare ad anomalie nel comportamento del sistema.
Sicurezza	Il sistema è relativamente sicuro, in quanto non sono necessarie connessioni ad internet e ad altre reti locali. Si accede al sistema attraverso login univoca con una relativa password in possesso di ogni utente registrato.

### 1.2.3 Maintenance Criteria

Criterio di desing	Definizione
Estendibilità	Il sistema è stato suddiviso in sottosistemi quasi del tutto indipendenti tra loro, quindi permette l'aggiunta di nuove funzionalità senza l'apporto di particolari modifiche.
Portabilità	Il sistema non presenta particolari problemi di portabilità poiché è realizzato in Java.
Adattabilità	Il sistema software non è adattabile poiché gestisce funzionalità create per soddisfare le richieste di un dominio applicativo specifico.
Leggibilità	Dalla lettura del codice, munito di commenti supportati da una documentazione più precisa sviluppata utilizzando il tool "javadoc", la comprensione del sistema risulta semplice grazie anche alla modularità dello stesso.
Tracciabilità dei requisiti	La tracciabilità dei requisiti è relativamente semplice in quanto c'è coerenza tra gli artefatti realizzati, sia per quanto riguarda la terminologia sia per i concetti contenuti.

### ***1.3 Definizioni, Acronimi e Abbreviazioni***

**DBMS:** Database Management System

**JDBC:** Java Database Connectivity

**SQL:** Structured Query Language

**RMI:** Remote Method Invocation

**MVC:** Model View Control

### ***1.4 Riferimenti***

- RAD – COFFEE **R**equirements **A**nalysis **D**ocument.

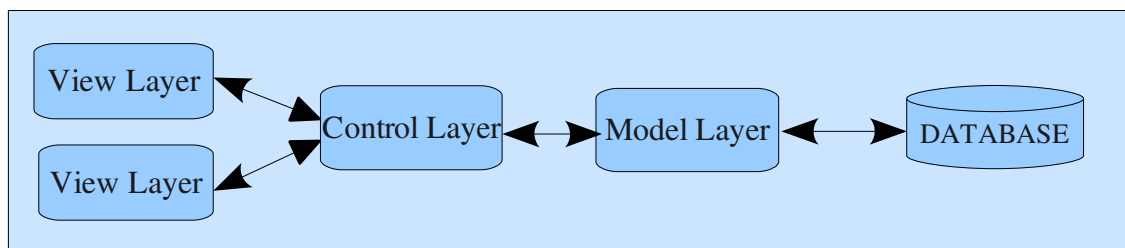
## 2. Architettura del sistema proposto

L'architettura del nostro sistema si presenterà suddivisa in tre diversi livelli seguendo lo schema dell'architettura MVC (Model/View/Control). Tale scelta è dovuta alla necessità di un'architettura che permetta una separazione netta tra i componenti software e i componenti che gestiscono i dati stessi.

Gli utenti di Coffee potranno accedere al sistema attraverso un client sul quale risiederanno l'interfaccia utente e la logica applicativa, mentre l'archivio dei dati e la sua gestione risiederanno su un unico server.

I diversi livelli previsti dall'architettura MVC sono:

- View Layer: il quale racchiuderà tutti gli oggetti boundary che costituiranno l'interfaccia grafica e attraverso i quali l'utente interagirà con il sistema.
- Control Layer: il quale conterrà tutti gli oggetti control che fungeranno da intermediari con le entità del sistema.
- Model Layer: nel quale saranno presenti tutti gli oggetti entity che si occuperanno



Architettura Model-Control-View



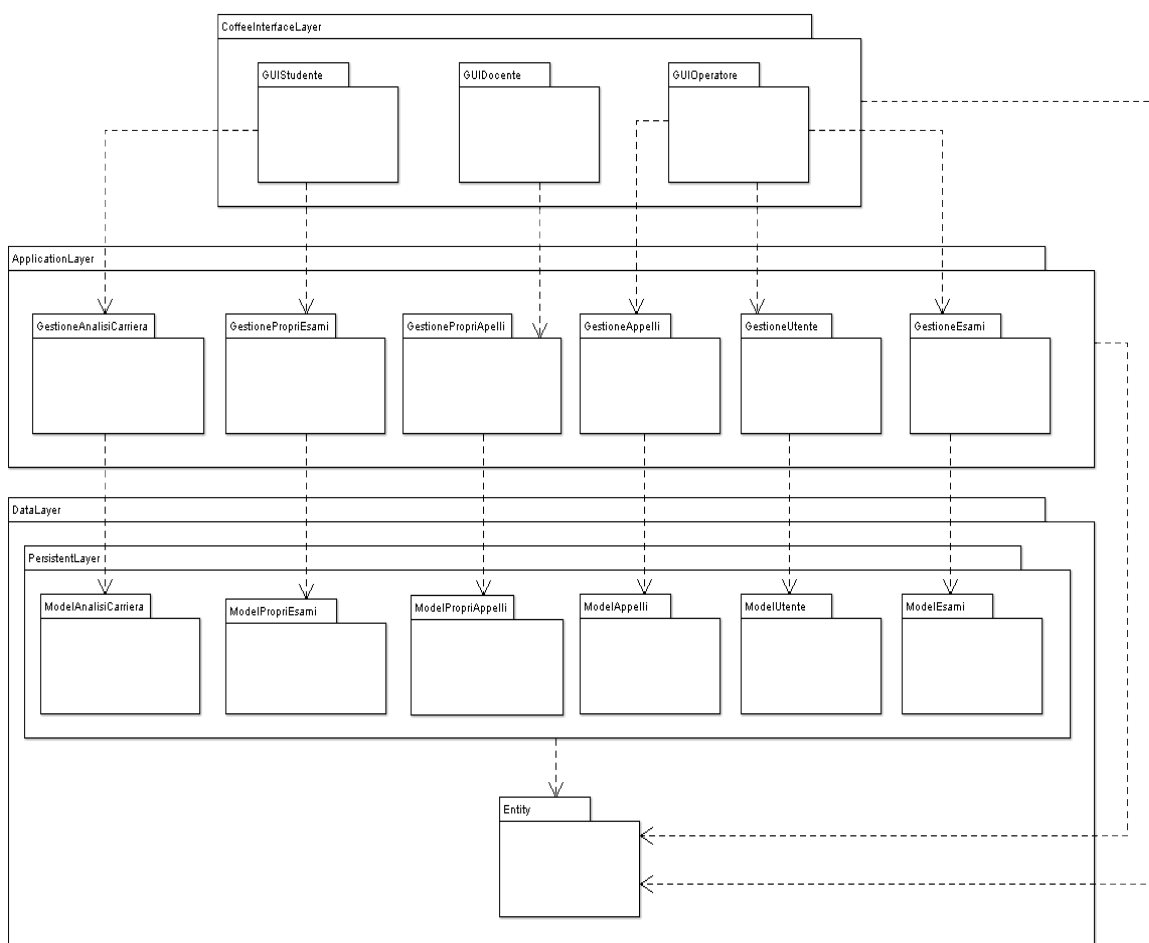
## 2.1 Subsystem decomposition

Ogni sottosistema è stato suddiviso in tre layer:

- CoffeeInterface Layer
- Application Layer
- Data Layer

L'utente interagirà con il *CoffeeInterfaceLayer*, il quale rappresenta l'interfaccia utente. Tale modulo interagirà con l'*ApplicationLayer* che eseguirà la funzionalità richieste ed, a sua volta, comunicherà con il *DataLayer* il quale permetterà, attraverso l'utilizzo di JDBC, l'interfacciamento con il DBMS.

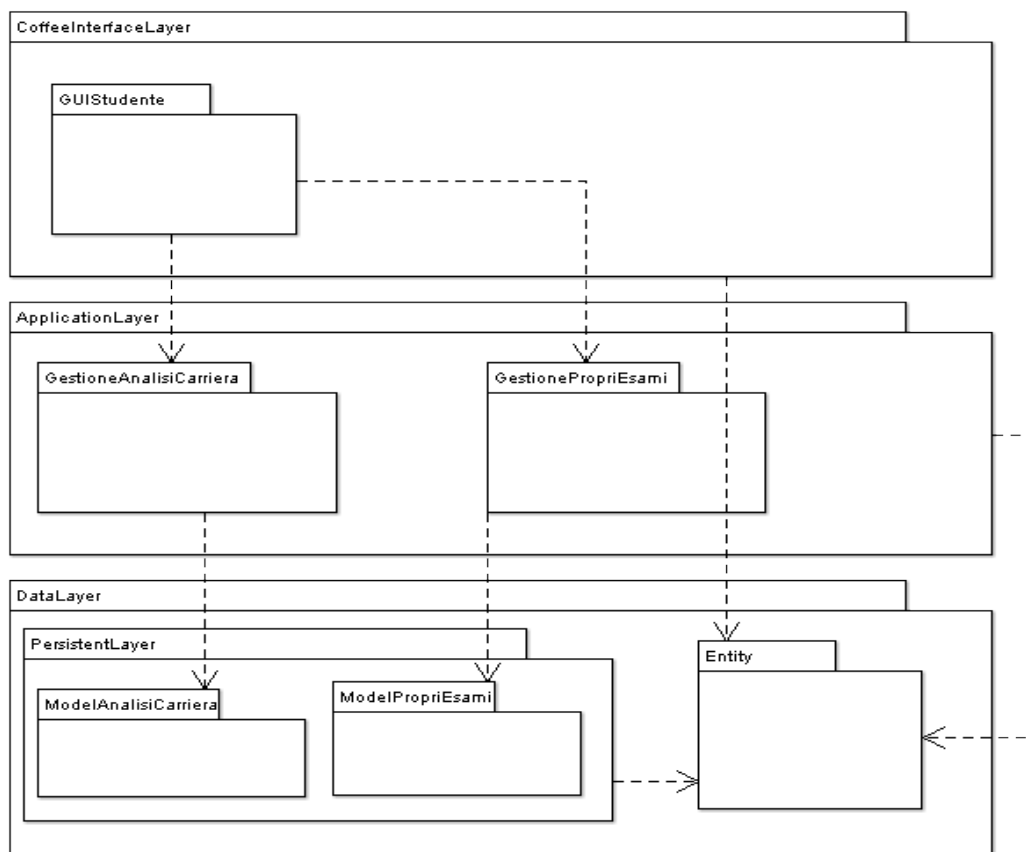
Di seguito è stata riportata una rappresentazione grafica globale della decomposizione in sottosistemi strutturata nei tre tre livelli precedentemente descritti:



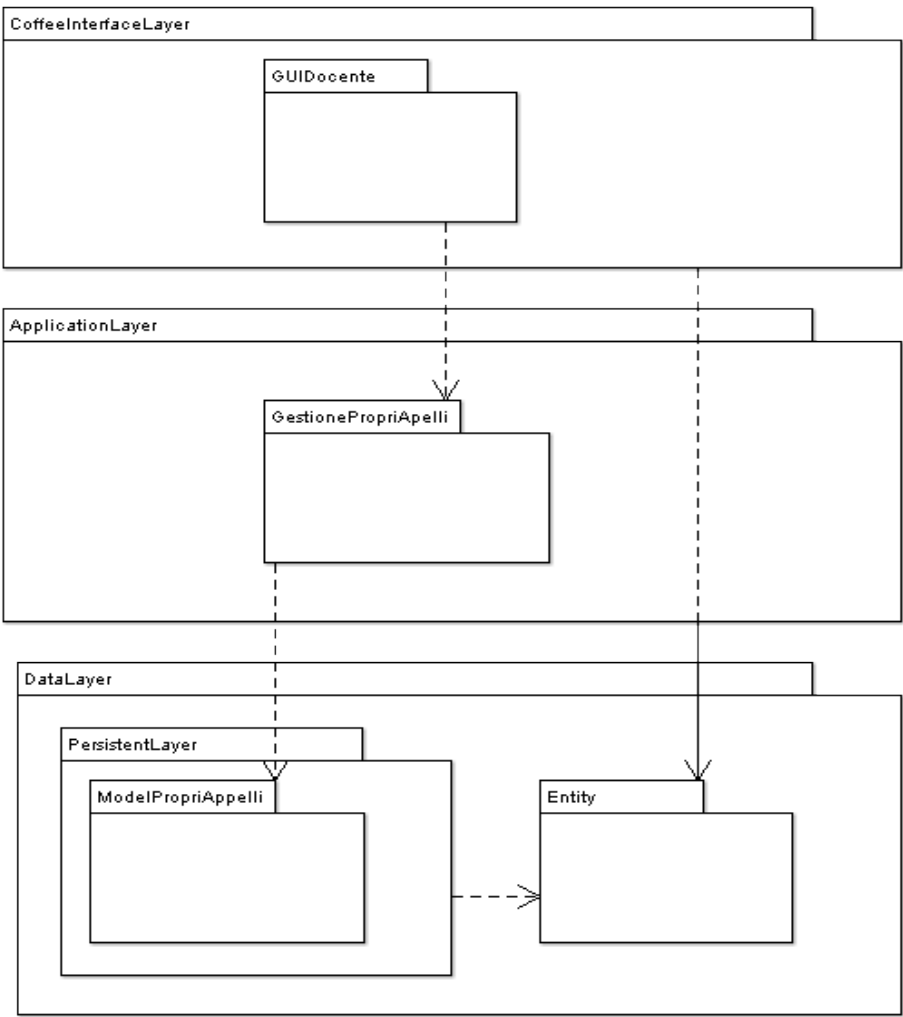
Considerando le funzionalità del sistema, sono stati individuati i fondamentali sottosistemi, ovvero:

- SottosistemaStudente
- SottosistemaDocente
- SottosistemaOperatore

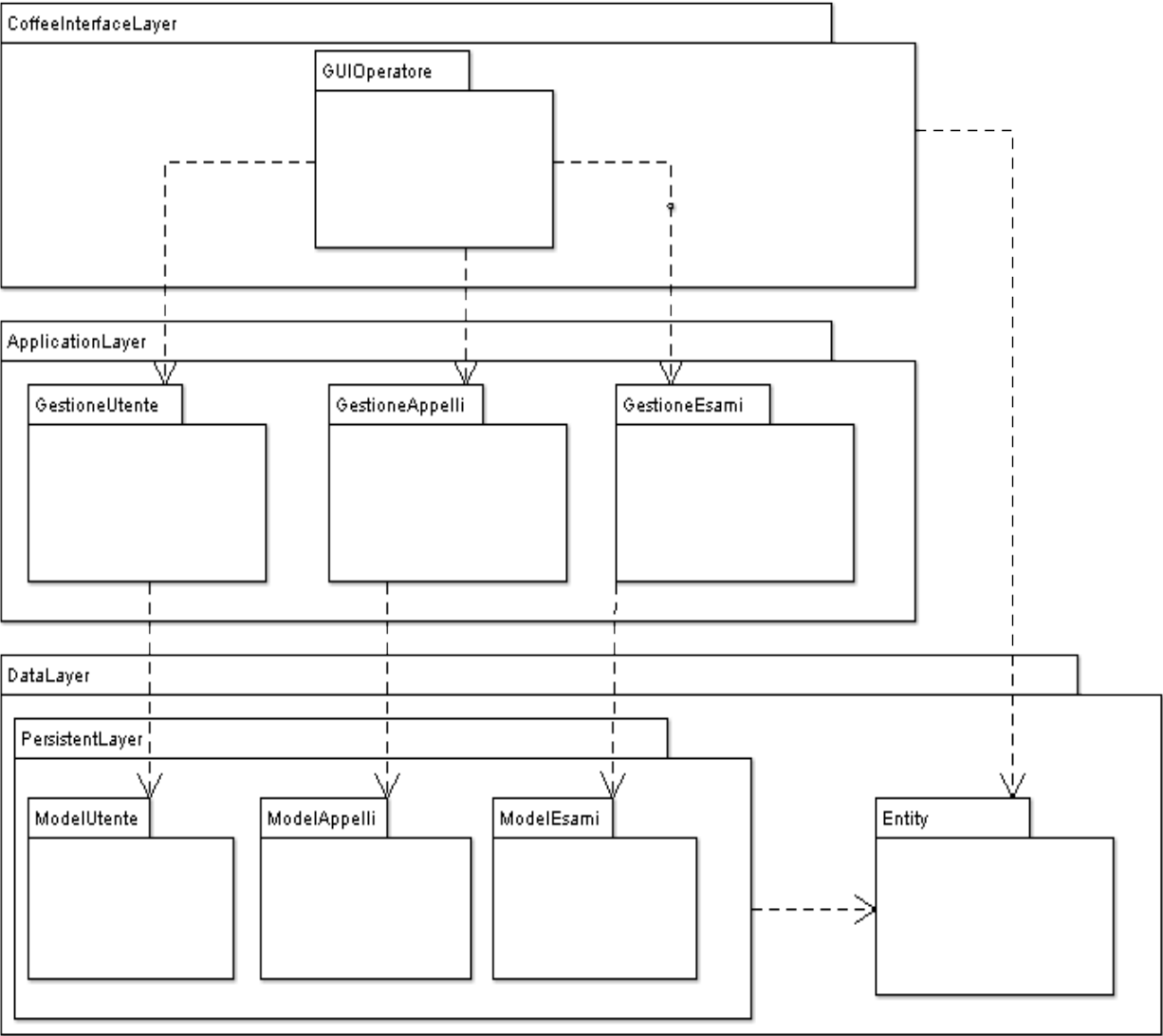
SottosistemaStudente:



SottosistemaDocente:



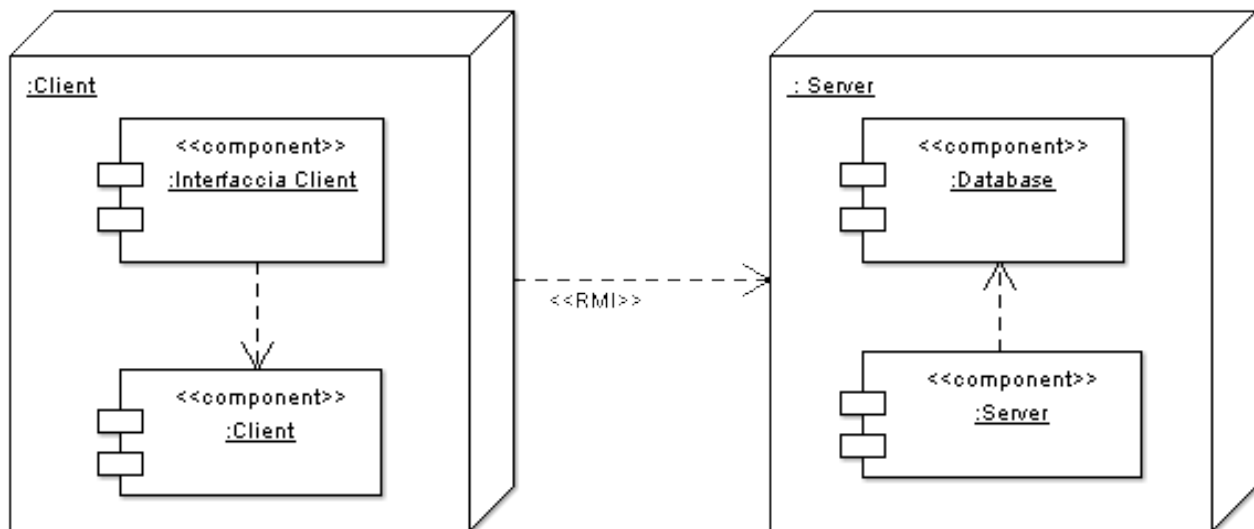
SottosistemaOperatore:



## 2.2 Hardware/Software mapping

L'architettura è stata scomposta ulteriormente in due livelli: Client e Server.

Il nodo Client è composto da *InterfacciaClient*, che fornisce l'interfaccia all'utente, e da



*Client*, che fornisce i metodi per la connessione al server.

Il nodo Server, invece, è composto da *Server*, il quale si occupa di gestire le richieste inviategli dall'utente tramite il client, e da *Database*, il quale si occupa di memorizzare i dati persistenti sul sistema.

I nodi comunicano tra loro attraverso la tecnologia RMI (Remote Method Invocation), mentre *Server* comunica con *Database* attraverso JDBC.

## ***2.3 Gestione dei Dati Persistenti***

Il sistema **COFFEE** presenta i seguenti oggetti da memorizzare in modo persistente:

- UtenteRegistrato:
  - Docente
  - Studente
  - Operatore
- Appello
- Libretto
- Prenotazione
- Aula
- Esame
- Richiesta

Per la memorizzazione dei dati è stato scelto un database relazionale in modo tale da memorizzare i dati in tabelle che seguono uno schema definito. L'organizzazione dei dati attraverso questo modello permette una semplice realizzazione di query complesse.

**UTENTE\_REGISTRATO**

Attributo	Descrizione
nomeUtente	E' la chiave primaria della tabella e rappresenta l'username che utilizza l'utente per accedere al sistema
password	Rappresenta la password che l'utente dovrà inserire oltre l'username per autenticarsi ed accedere al sistema.
nome	Costituisce il nome dell'utente
cognome	Costituisce il cognome dell'utente
dataNascita	Rappresenta la data di nascita dell'utente
indirizzo	Rappresenta l'indirizzo di residenza dell'utente

**DOCENTE:**

Attributo	Descrizione
matricola	E' la chiave primaria della tabella e rappresenta un identificativo del docente
tipo_contratto	Descrive il tipo di contratto con cui il docente è stato assunto
nomeUtente	E' la chiave esterna che identifica il docente come utente registrato

**STUDENTE:**

Attributo	Descrizione
matricola	E' la chiave primaria della tabella e rappresenta l'identificativo dello studente
anno_iscrizione	Rappresenta l'anno di iscrizione presso l'università dello studente
nomeUtente	E' la chiave esterna che identifica lo studente come utente registrato

**OPERATORE:**

Attributo	Descrizione
nomeUtente	Funge sia da chiave esterna sia da chiave primaria e rappresenta l'identificativo dell'operatore
data_registrazione	Rappresenta la data in cui è stato registrato l'operatore

**ESAME:**

Attributo	Descrizione
nome	E' la chiave primaria della tabella e rappresenta il nome dell'esame
ore_laboratorio	Rappresenta le ore di laboratorio che prevede l'esame cui si riferisce
ore_frontali	Costituisce le ore di lezioni frontali che prevede l'esame cui si riferisce
crediti	Rappresenta il numero di crediti formativi associati all'esame
programma	Costituisce Un file pdf contenente il programma del corso

**LIBRETTO:**

Attributo	Descrizione
matr_studente	La matricola dello studente
♦ esame	Il nome dell'esame da esso sostenuto
(matr_studente,esame)	Chiave primaria della tabella
voto	Costituisce il voto con cui lo studente ha superato l'esame
data	Rappresenta la data in cui è stato sostenuto l'esame



**PRENOTAZIONE:**

Attributo	Descrizione
codice	E' la chiave primaria della tabella che viene generata automaticamente ogni volta che viene effettuata una prenotazione
matr_studente	E' la chiave esterna che rappresenta lo studente cui si riferisce la prenotazione
appello	E' la chiave esterna che rappresenta l'appello a cui si è prenotato lo studente
data_prenotazione	costituisce la data in cui viene effettuata la prenotazione all'esame

**AULA:**

Attributo	Descrizione
nominativo	E' la chiave primaria della tabella e identifica univocamente l'aula
num_posti	Rappresenta la capacità di posti dell'aula.
ubicazione	Costituisce un'indicazione circa l'ubicazione dell'aula, ad esempio il piano e l'edificio in cui si trova.
Disponibilità	Campo che indica se l'aula è libera o meno.

**RICHIESTA:**

Attributo	Descrizione
num_prog	è la chiave primaria della tabella e rappresenta l'identificativo della richiesta generato automaticamente ogni volta che una richiesta viene inviata
matr_docente	è la chiave esterna che rappresenta il docente che ha inviato la richiesta
data	rappresenta la data di invio della richiesta
oggetto	costituisce l'oggetto della richiesta
testo	contiene il file di testo della richiesta

**APPELLO:**

Attributo	Descrizione
matr_docente	La matricola del docente che ha stabilito l'appello
esame	Il nome dell'esame a cui fa riferimento l'appello
data	La data cui si potrà sostenere l'appello
(matr_docente, esame, data)	Chiave primaria della tabella
tipologia	Rappresenta il tipo d'esame(scritto-orale-pratico)
ora_inizio	Costituisce l'ora di inizio dell'esame
durata	Rappresenta l'eventuale durata dell'esame(nel caso in cui la tipologia sia scritto o pratico)
♦ vincoli	Rappresenta eventuali vincoli da rispettare per sostenere l'esame

## ***2.4 Controllo degli accessi e sicurezza***

### **2.4.1 Controllo degli accessi**

Il sistema software COFFEE prevede, in generale, tre figure di utenti: Amministratore, Utente Guest e Utente Registrato. A sua volta, Utente Registrato, si specializza in tre figure di utenti distinte: Studente, Docente e Operatore. Ogni utente accede, tramite procedura di login, a sottosistemi diversi, i quali permettono di usufruire delle funzionalità specifiche ad essi associate.

Per tenere traccia dei diritti di accesso, all'interno del sistema si usufruisce di una tabella di controllo degli accessi che descrive per ogni attore a quali oggetti gli è consentito accedere. Tale tabella può essere vista come una matrice, dove le righe rappresentano gli attori e le colonne rappresentano gli oggetti su cui sono regolamentati gli accessi.

Nel progetto in questione gli **attori** sono:

- ◆ Amministratore
- ◆ Utente Guest
- ◆ Studente
- ◆ Docente
- ◆ Operatore

(L'attore in precedenza menzionato, UtenteRegistrato, non è stato incluso in quanto le relative specializzazioni presentano diritti di accesso diversi).

Gli **oggetti** sono:

- ◆ libretto
- ◆ appello
- ◆ esame
- ◆ prenotazione

- ♦ aula
- ♦ richiesta
- ♦ docente
- ♦ studente

<b>Oggetti</b> <b>Attori</b>	<i>Libretto</i>	<i>Appello</i>	<i>Esame</i>
<i>Operatore</i>	<<create>> RegistraEsame ModificaRegistrazione	<<create>> AggiungiAppello ModificaAppello RicercaAppello VisualizzaAppello	<<create>> AggiungiEsame ModificaEsame RicercaEsame VisualizzaEsame
<i>Studente</i>	VisualizzaLibretto		
<i>Docente</i>		RichiediAppello ModificaRichiesta Appello	

<b>Oggetti</b>	<i>Prenotazione</i>	<i>Comunicazione</i>	<i>UtenteRegistrato</i>
<b>Attori</b>			
<i>Studente</i>	<<create>> AggiungiPrenotazione		
<i>Docente</i>		<<create>> RichiediAppello EliminaAppello RichiediModifica Appello	
<i>Utente Guest</i>			<<create>> RichiediAccount

## 2.4.2 Sicurezza

Il sistema COFFEE è protetto dagli accessi non autorizzati attraverso un sistema login/password. Ogni utente registrato al sistema è proprio di una login univoca ed una password di minimo 6 caratteri e massimo 10. Un utente per poter avere accesso al sistema deve compilare due campi: login e password. Il sistema quindi analizza i campi login e password e controlla l'esistenza di una tupla con gli specifici dati inviati all'interno del database. Nel caso in cui la tupla ricercata non fosse presente nel database viene generato un messaggio di errore e riproposto l'inserimento dei dati.

Il modello login/password utilizzato non fornisce sicuramente un elevato livello di sicurezza, ma considerando che gli accessi possono essere effettuati solo sulla rete locale, il livello di sicurezza viene considerato sufficiente per il contesto dell'applicazione.

## ***2.5 Gestione del controllo globale***

Il flusso di controllo di questo sistema software non prevede alcuna sequenza prestabilita. Sarà l'iterazione con l'utente a determinare il flusso di controllo, le classi Java funzioneranno da ricevitori di eventi per rispondere alle attività dei client. Essendo l'applicazione implementata con tecnologia RMI (Remote Method Invocation), ogni qualvolta l'utente vorrà eseguire delle operazioni, dovrà interagire con un'interfaccia attraverso la quale richiederà dei servizi. La richiesta partirà dal client, che è collegato con uno stub, il quale invierà un segnale allo skeleton con cui è in comunicazione. Lo stesso skeleton, ricevuto il segnale, impartirà delle richieste al server, con il quale è collegato. Elaborata la richiesta, il server prenderà le informazioni dal database con il quale comunica. Queste informazioni contenute nel database, faranno il percorso inverso fino ad arrivare all'utente. Per primo le prenderà il server, che grazie allo skeleton le invierà allo stub. Lo stub le riceverà e le comunicherà al client e grazie ad un'interfaccia grafica saranno visualizzate dall'utente.

## **2.6 Boundary conditions**

### **2.6.1 Configuration**

Per ogni oggetto persistente ora verranno definite le fasi in cui vengono creati, distrutti e archiviati nel database.

**UtenteRegistrato:** l'oggetto viene creato dall'Utente Guest seguendo il caso d'uso RichiediAccount e può essere eliminato dall'Amministratore o da Operatore seguendo il caso d'uso EliminaAccount.

**Appello:** l'oggetto viene creato da Operatore seguendo il caso d'uso AggiungiAppello e può essere rimosso dallo stesso seguendo il caso d'uso EliminaAppello.

**Libretto:** l'oggetto viene creato da Operatore seguendo il caso d'uso RegistraEsame e archiviato all'uscita dello stesso. La rimozione dal sistema può essere effettuata solo dall'Amministratore del sistema.

**Prenotazione:** l'oggetto viene creato da Studente seguendo il caso d'uso PrenotaEsame e archiviato all'uscita del medesimo. La rimozione può essere effettuata dallo stesso seguendo un altro caso d'uso, EliminaPrenotazione, oppure dall'Amministratore.

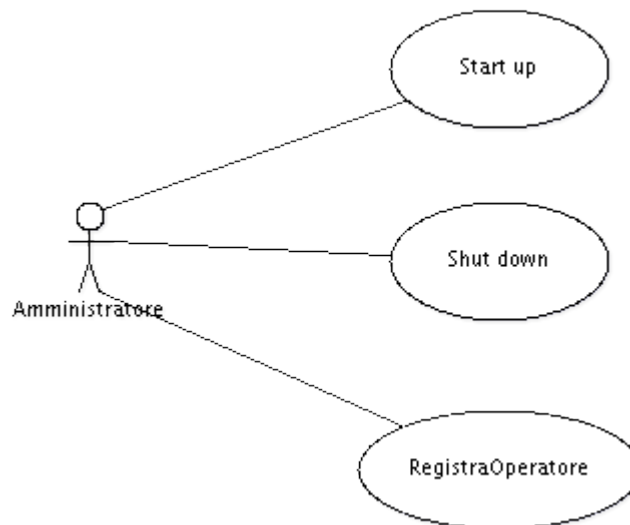
**Aula:** l'oggetto viene creato e distrutto dall'Amministratore oppure da Operatore seguendo rispettivamente i caso d'uso CreaAula e EliminaAula. L'archiviazione dello stesso avviene all'uscita del caso d'uso CreaAula.

**Esame:** l'oggetto viene creato e distrutto dall'Operatore tramite i casi d'uso creaEsame ed eliminaEsame.

**Richiesta:** l'oggetto viene creato da Docente all'uscita dei casi d'uso che vengono utilizzati per richiedere operazioni all'operatore, quali RichiediAggiuntaAppello e RichiediModificaDataAppello, e può essere rimosso sia da Operatore che dall'Amministratore.



## 2.6.2 Start up – Shut down use cases



### 2.6.2.1 Start Up

<b>Use Case Name</b>	<b>Start Up</b> <i>L'Amministratore effettua lo Start UP del server.</i>
<b>Participating Actors</b>	<ul style="list-style-type: none"> <li>• Inizializzato da <b><u>Amministratore</u></b></li> </ul>
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>• L'Amministratore ha accesso fisico al sistema.</li> </ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>1. L'Amministratore accede alla macchina server.</li> <li>2. Il sistema autentica l'Amministratore e lo fa accedere alla shell dei comandi.</li> <li>3. L'Amministratore avvia il server.</li> <li>4. Il Sistema conferma l'avvenuta operazione.</li> <li>5. Il Server è attivo.</li> </ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>• Il server è stato attivato.</li> </ul>
<b>Quality Requirements</b>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>

**2.6.2.2 Shut down**

<b>Use Case Name</b>	<b>Shut down</b> <i>L'Amministratore effettua lo shut down del server.</i>
<b>Participating Actors</b>	Inizializzato da <b><u>Amministratore</u></b>
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>● L'Amministratore ha accesso fisico al sistema.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. L'Amministratore accede alla macchina server.</li><li>2. Il sistema autentica l'Amministratore e lo fa accedere alla shell dei comandi.</li><li>3. L'Amministratore termina il server.</li><li>4. Il sistema porta a termine le operazioni in corso per garantire uno stato di coerenza dei dati.</li></ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>● Il server è stato disattivato.</li></ul>
<b>Quality Requirements</b>	<ul style="list-style-type: none"><li>● N/A</li></ul>

**2.6.2.3 RegistraOperatore:**

<b>Use Case Name</b>	<b>RegistraOperatore</b> <i>L'Amministratore registra un nuovo operatore al sistema.</i>
<b>Participating Actors</b>	<ul style="list-style-type: none"><li>● Inizializzato da <u><b>Amministratore</b></u></li></ul>
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>● L'amministratore ha accesso fisico al sistema.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. L'Amministratore accede alla macchina server.</li><li>2. Il sistema autentica l'Amministratore e lo fa accedere alla shell dei comandi.</li><li>3. L'Amministratore aggiunge un operatore al sistema, associandogli un NomeUtente univoco ed una password.</li><li>4. Il sistema conferma l'operazione.</li></ol>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>● L'operatore è stato aggiunto.</li></ul>
<b>Quality Requirements</b>	<ul style="list-style-type: none"><li>● Non può essere aggiunto un attore già presente nel sistema.</li></ul>

**2.6.3 Exception handling**

1. Server disconnesso da rete o in crash: il client visualizzerà un messaggio di errore che comunica l'impossibilità di contattare il Server e tutti i servizi del sistema non saranno disponibili fino a quando il problema non sarà risolto.

### 3. Servizi dei Sottosistemi

Descriviamo ora i servizi forniti da ognuno dei sottosistemi in termini di operazioni.

<b>Sottosistema</b>	<i>AnalisiCarriera</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni di analisi della carriera degli studenti
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
VisualizzaLibretto	Il servizio permette ad uno studente di visualizzare il proprio libretto
VisualizzaStatistiche	Il servizio permette ad uno studente di visualizzare alcune statistiche sulla sua carriera universitaria

<b>Sottosistema</b>	<i>GestioneUtenti</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni sugli account.
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
RichiediAccount	Il servizio permette di richiedere un nuovo account per utilizzare il sistema.
RichiediModificaAccount	Il servizio permette di modificare un account già esistente nel database. Prima viene letto l'account, poi viene modificato ed infine sovrascritto nel database
EliminaAccount	Il servizio permette di rimuovere permanentemente un account dal database

<b>Sottosistema</b>	<i>GestioneAppelli</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni sugli appelli.
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
RegistraPrenotazioneAppello	Il servizio permette all'operatore di registrare una prenotazione ad un appello da parte di uno studente, inserendola nel database
AggiungiAppello	Il servizio permette all'operatore di aggiungere un nuovo appello al database, in seguito ad una richiesta di un docente
ModificaDataAppello	Il servizio permette all'operatore di modificare i dati di un appello già registrato nel database. L'appello viene prima letto, poi modificato e quindi sovrascritto nel database
EliminaAppello	Il servizio permette all'operatore di eliminare un appello dal database

<b>Sottosistema</b>	<i>GestioneEsame</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni sugli esami
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
CreaEsame	Il servizio permette all'operatore di inserire un nuovo esame all'interno del database
ModificaEsame	Il servizio permette all'operatore di modificare un esame presente nel database.
RimuoviEsame	Il servizio permette all'operatore di eliminare un esame dal database.

<b>Sottosistema</b>	<i>GestioneSessioneDiLavoro</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni sugli account
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Login	Il servizio permette l'accesso al sistema
Logout	Il servizio permette di disconnettere dal sistema un UtenteRegistrato che ha fatto accesso al sistema
ModificaPassword	Il servizio che permette di modificare la password di accesso
RecuperaPassword	Il servizio permette di recuperare la password di un account

<b>Sottosistema</b>	<i>GestionePropriAppelli</i>
<b>Descrizione</b>	Sottosistema che gestisce le operazioni sugli appelli
<b>Servizi offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
RichiediAggiunta Appello	Il servizio permette ad un Docente di richiedere l'aggiunta di un appello
RichiediModificaDataAppello	Il servizio permette ad un Docente di richiedere la modifica dei dati di un appello registrato
RegistraVoto	Il servizio permette ad un Docente di registrare il voto dell'esame di uno studente prenotato che ha superato l'esame
RegistraRitiro	Il servizio permette ad un Docente di registrare il ritiro dell'esame di uno studente prenotato
RegistraAssenti	Il servizio permette ad un Docente di registrare l'assenza dall'esame di uno studente prenotato
RicercaPrenotati	Il servizio permette di visualizzare tutti i prenotati ad un appello

In allegato si inserisce una prima bozza del database.

