



UNIVERSITÀ DEGLI STUDI DI SALERNO

Ingegneria del Software

SYSTEM DESIGN DOCUMENT



ANNO ACCADEMICO 2018/2019

TOP MANAGER:

Nome
Prof. Andrea De Lucia

PARTECIPANTI:

Nome	Matricola
Mario Sessa	0512104650
Luigi Crisci	0512104740
Pasquale Ambrosio	0512104704

HISTORY:

Data	Versione	Cambiamenti	Autori
15/12/2018	1.0	Stesura delle componenti	Luigi Crisci
23/12/2018	1.1	Inserimento sezioni 1, 2 e formattazione del documento	Luigi Crisci, Mario Sessa
26/12/2018	1.2	Completamento SDD.	Luigi Crisci
26/12/2018	1.3	Formattazione parziale del documento	Mario Sessa
27/12/2018	1.4	Formattazione completa del documento, correzione sezione 2.4	Mario Sessa
29/12/2018	1.4.1	Modifica sezione 2.4	Luigi Crisci
14/02/2019	1.5	Correzioni logiche EER e servizi dei sottosistemi	Mario Sessa
16/02/2019	2.0	Supervisione documento	Mario Sessa,

			Pasquale Ambrosio
--	--	--	----------------------

Table of Contents

1. INTRODUZIONE	7
1.1 SCOPO DEL SISTEMA	7
1.2 OBIETTIVI DI DESIGN	7
1.2.1 CRITERI DI PERFORMANCE	7
1.2.2 CRITERI DI AFFIDABILITÀ	8
1.2.3 CRITERI DI MANUTENIBILITÀ	8
1.2.4 CRITERI PER L'UTENTE FINALE	8
1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI	8
1.4 RIFERIMENTI	9
1.5 PANORAMICA	9
2. ARCHITETTURA DEL SISTEMA PROPOSTO	10
2.1 PANORAMICA	10
2.2 DECOMPOSIZIONE IN SOTTOSISTEMI	11
2.3 MAPPING HARDWARE-SOFTWARE	13
2.3.1 WEB SERVER	13
2.3.2 INTERFACE LAYER	13
2.3.3 APPLICATION LOGIC LAYER	13
2.3.4 STORAGE LAYER	13
2.3.5 DATABASE SERVER	14
2.4 GESTIONE DATI PERSISTENTI	14
2.4.1 ACCOUNT	15
2.4.3 ISCRIZIONE	16
2.4.4 CARTA DI CREDITO	16
2.4.5 LEZIONE	17
2.4.6 COMMENTO	17
2.5 CONTROLLO DEGLI ACCESSI E SICUREZZA	18
2.6 CONTROLLO GLOBALE DEL SOFTWARE	19
2.7 CONDIZIONE DI BOUNDARY	19
[UC1] UC_STARTUP	19
[UC2] UC_SHUTDOWN	20
2.7.1 FALLIMENTO	21

3.	<i>SERVIZI DEI SOTTOSISTEMI</i>	22
3.1	GESTIONE CORSI	22
3.2	GESTIONE LEZIONI:	23
3.3	GESTIONE PAGAMENTO:	24
3.4	GESTIONE MAIL:	24
3.5	GESTIONE UTENTE:	25
4.	GLOSSARIO	26

1. Introduzione

1.1 Scopo del sistema

Gli obiettivi principali del sistema sono quelli di realizzare un software che permetta di creare e/o seguire dei corsi formativi online. Il progetto "YouLearn" propone quindi una piattaforma online dove gli utenti potranno scegliere se essere studenti, ed iniziare fin da subito ad acquistare e seguire corsi, oppure docenti e creare i propri corsi. Vediamo che entrambi i ruoli possono appartenere ad un singolo account, inoltre non bisogna attuare nessun meccanismo di scambio di viste o ruoli dell'utente dato che la vista di docente e studente variano a secondo della semantica della pagina che si visualizza.

La pubblicazione di un corso nel sistema sarà permessa solo se approvato da un Supervisore, il quale si occuperà, per un qualunque corso creato e confermato da un docente, di visionarlo alla ricerca di contenuti non appropriati.

Il software si propone di essere funzionale sia su desktop che su dispositivi mobili come tablet e smartphone.

Inoltre, uno dei punti fermi del progetto sarà quella di implementare un'interfaccia di facile comprensione da parte della clientela così da fornire un sistema deduttivo e facile nel suo utilizzo.

1.2 Obiettivi di design

Il sistema dovrà poter essere il più efficiente ed intuitivo possibile. Tale efficienza sarà data da brevi tempi di risposta ad ogni genere di input. Si punterà ad aver una buona manutenibilità attraverso il facile inserimento di nuove funzionalità dovuto ad una strutturazione del software semplice da manipolare. Altro aspetto da considerare sono le rigide politiche che eviteranno che l'utente sbagli ad inserire alcuni input.

Parte degli aspetti che il sistema avrà sono:

1.2.1 Criteri di performance

Criteri:	Descrizione:
Tempo di risposta:	YouLearn assicura una risposta rapida alle richieste dell'utente, la quale verrà gestita ed elaborata entro 10 secondi. Nel caso di una connessione molto lenta tale aspetto potrebbe essere non garantito, ovviamente i tempi di risposta saranno più rapidi quanto più è veloce la tipologia di connessione utilizzata.
Memoria:	La quantità di memoria che verrà utilizzata dal sito web non può essere stimata precisamente. Il sistema potrebbe essere sottoposto alla memorizzazione di centinaia di corsi comprensivi di decine di lezione e migliaia di utenti, quindi una buona scalabilità in tal senso dovrà essere garantita.

1.2.2 Criteri di affidabilità

Criteri:	Descrizione:
Robustezza:	YouLearn gestirà eventuali input errati senza interrompere il funzionamento dell'intero sistema, l'utente verrà avvertito con eventuali messaggi di errore.
Sicurezza:	L'accesso al sistema è controllato da un sistema di autenticazione che categorizza gli utenti non permettendo l'accesso a parti del sistema non autorizzate, oltre alla presenza di filtri che impediranno accessi non previsti.

1.2.3 Criteri di manutenibilità

Criteri:	Descrizione:
Estendibilità:	Il sistema dovrà essere propenso all'inserimento di nuove funzionalità. Quindi, si adotterà una metodologia di programmazione ben strutturata, dunque, di facile comprensione.
Modificabilità:	L'accesso al sistema è controllato da un sistema di autenticazione che categorizza gli utenti non permettendo l'accesso a parti del sistema non autorizzate, oltre alla presenza di filtri che impediranno accessi non previsti.

1.2.4 Criteri per l'utente finale

Criteri:	Descrizione:
Usabilità:	YouLearn sarà di semplice utilizzo, grazie ad una grafica semplice ed efficace che guiderà gli studenti nell'acquisto dei corsi ed i docenti nella creazione corretta degli stessi.

1.3 Definizioni, acronimi e abbreviazioni

YouLearn: Nome del sistema che verrà sviluppato.

JDBC: Java Database Connection.

DBMS: Database Management System, Sistema di gestione del database.

MySQL: è il più diffuso DBMS relazionale open source in circolazione.

RAD: Requirements Analysis Document.

SDD: System Design Document.

User-friendly: aggettivo utilizzato per definire un software di facile utilizzo anche per persone non esperte nell'uso del PC.

Supervisore: Amministratore del sistema in grado di poter rendere pubblico o meno un corso in fase di attesa.

Studente: Astrazione semantica di un account quando visita, segue o si iscrive ad un corso.

Docente: Astrazione semantica di un account quando crea o gestisce un corso.

1.4 Riferimenti

Il contesto è ripreso dal documento di analisi dei requisiti del progetto YouLearn.

È stato anche usato come riferimento il libro:

Object-Oriented Software Engineering:

Using UML, Patterns, and Java, 3rd Edition

Prentice Hall, Upper Saddle River, NJ, September 25, 2009.

Infine, sono stati usati dei materiali di supporto visionabili al link:

https://www.bruegge.in.tum.de/lehstuhl_1/component/content/article/217-OOSE

1.5 Panoramica

In questa sezione presentiamo una panoramica di ciò che riguarderà il documento. Questa forma di indicizzazione serve per orientare il lettore su come si struttura delle fondamenta dell'architettura software che si sta sviluppando così da poter essere il più chiari possibili. Di seguito riportiamo le varie parti del contesto che verrà analizzato:

1. **Decomposizione del sistema:** Il sistema viene suddiviso in diversi sottosistemi. Ogni sottosistema è caratterizzato da servizi che offre ad altri sottosistemi. L'insieme dei servizi sarà denominato Interfaccia.

2. **Mapping Hardware/Software:** in questa sezione vengono prese decisioni riguardo le piattaforme hardware scelte per il sistema, la comunicazione tra nodi, e come vengano incapsulati i servizi di un sottosistema.

3. **Gestione dei dati persistenti:** In tale sezione vengono individuati gli oggetti che devono essere resi persistenti e quale software verrà utilizzato per tale scopo.

4. **Politiche di accesso e Sicurezza:** In tale sezione si individueranno grazie ad una tabella le operazioni che ogni attore può effettuare.

5. **Controllo del software globale:** che descrive il modo in cui è implementato il controllo globale del software e come si sincronizzano i sottosistemi.

6. **Condizioni Boundary:** che descriverà oltre l'avvio anche la gestione dei fallimenti, interruzione di corrente o anche a errori di progettazione.

2. Architettura del Sistema Proposto

2.1 Panoramica

Il sistema che intendiamo sviluppare è un software web-based e sarà un sistema dove sarà possibile pubblicare dei corsi e seguirne, previo il loro acquisto.

Le tipologie di utenti si dividono in: Studente, Docente e Supervisore.

L'utente non registrato non potrà accedere a nessuna delle funzionalità del sistema, se non la registrazione.

Lo Studente potrà accedere alle funzionalità che permettono l'acquisto di un corso, la visione dei contenuti del corso e l'inserimento di commenti alle lezioni. Potrà inoltre visualizzare e modificare le proprie informazioni personali

Il docente potrà accedere alle funzionalità che permettono la creazione di un corso, la visione dei contenuti del corso e la gestione dei commenti delle lezioni dei propri corsi. Potrà inoltre visualizzare e modificare le proprie informazioni personali.

Infine, il Supervisore si occuperà, per ogni corso appena creato da un docente, di controllarne la qualità e di accettarlo, se ritenuto adatto, oppure di rifiutarlo dando feedback sul motivo del rifiuto.

L'architettura del sistema sarà quindi di tipo Client-Server.

Il server riceverà le richieste da parte del Client e cercherà di rispondere in un tempo ristretto.

I motivi di questa scelta sono:

1. **Portabilità:** Essendo un'applicazione web, potrà essere usata su una varietà di macchine e sistemi operativi e, quindi, sia da computer fissi che dispositivi mobili.
2. **Scalabilità:** il server sarà in grado di gestire un grosso numero di richieste da parte dei client.
3. **Affidabilità:** entrambe le componenti Client-Server garantiranno affidabilità sui dati e, quindi, anche in seguito a guasti sarà possibile recuperare le informazioni. Per tale motivo verranno effettuati backup periodici al database con cadenza mensile.
4. **Performance:** Il sistema garantirà reattività nel completamento delle task, anche se tale aspetto è molto influenzato dalla qualità della connessione ad internet che l'utente finale possiede.

Nello sviluppo del software utilizzeremo un'architettura di tipo MVC, tale pattern prevede che il software venga diviso in tre parti ognuna delle quali ha un compito diverso.

La view si occuperà di curare l'interazione con l'utente, Il controller riceverà informazioni della View per interfacciarsi al model, mentre il model si occuperà della gestione dei dati e, quindi, dell'iterazione con il database

sottostante. L'utilizzo di questo modello comporta numerosi vantaggi, primi tra tutti sono la

facilità di manutenzione ed estensione. Infatti, essendo uno standard di programmazione, sarà di facile comprensione a programmatori futuri.

I model verranno realizzati utilizzando classi Java appropriate, la parte di view verrà implementata utilizzando HTML e Java in pagine JSP e i control saranno realizzati tramite Servlet.

2.2 Decomposizione in sottosistemi

Per la realizzazione del sistema sarà utilizzata una architettura three-tier. Utilizzeremo una particolare architettura multi-tier in cui la logica dell'applicazione verrà suddivisa in tre parti detti layer:

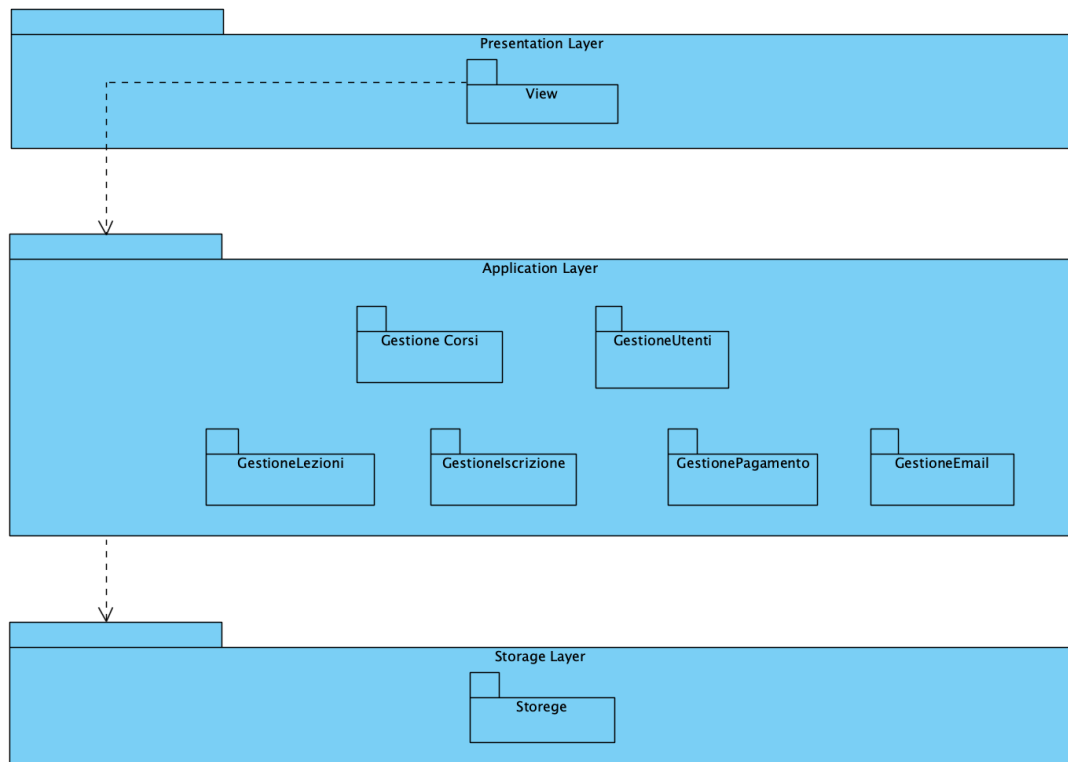
1. Presentation layer: composto da tutte le interfacce grafiche, in particolare dai boundary object come form che verranno compilati dagli utenti.
2. Application layer: composta dagli oggetti che si occuperanno dell'elaborazione dati e di notificare cambiamenti al presentation layer. Inoltre, questo strato interagirà con il database sottostante tramite lo storage layer.
3. Storage layer: Si occupa della memorizzazione di dati persistenti e del loro recupero dal database attraverso l'uso di query.

Il sottosistema del presentation layer è stato suddiviso in quattro sottosistemi

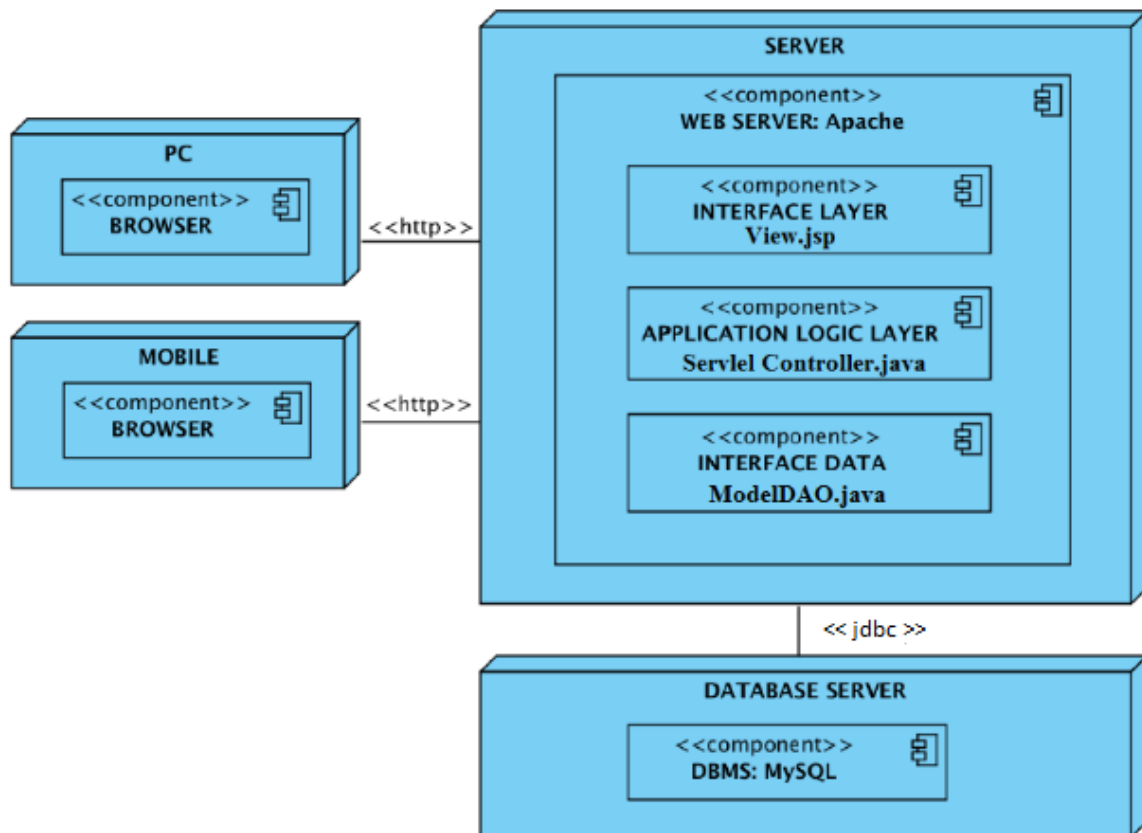
1. Sottosistema Autenticazione: Contiene le pagine di benvenuto.
2. Sottosistema Studente: oltre a tutte le interfacce grafiche del sottosistema Autenticazione, potrà accedere alla funzionalità di acquistare corsi, alla funzionalità di ricerca e di inserimento commenti e\o cancellazione (solo quelli da lui inseriti). Inoltre, potrà accedere ad interfacce grafiche aggiuntive come quella per la visualizzazione delle informazioni personali e la loro modifica.
3. Sottosistema Docente: Tale modulo si occuperà di gestire le funzionalità di creazione di un corso, visualizzazione dei contenuti dei propri corsi e gestione dei commenti delle proprie lezioni. Inoltre, potrà accedere ad interfacce grafiche aggiuntive come quella per la visualizzazione delle informazioni personali e la loro modifica.
4. Sottosistema Supervisore: include le interfacce accessibile al supervisore per verifica dei corsi creati dai docenti del sistema.

Il sottosistema application layer è stato suddiviso in vari sottosistemi

1. Sottosistema Gestione corsi: comprende tutte le operazioni per la creazione, la modifica, la verifica di un corso
2. Sottosistema Gestione lezioni: comprende tutte le operazioni per l'inserimento e la visualizzazione di una lezione, l'aggiunta e la rimozione dei commenti;
3. Sottosistema Gestione utenti che include tutte le operazioni di creazione di un utente, visualizzazione delle informazioni personali, eliminazione di un utente e la verifica dei dati in fase di Login.
4. Sottosistema Iscrizione: gestisce l'iscrizione di uno studente ad un corso
5. Sottosistema Gestione Pagamento: include tutte le operazioni per il completamento della transazione per l'acquisto di un corso (servizio esterno)
6. Sottosistema Gestione E-mail: include le operazioni per l'invio di e-mail di notifica agli utenti del sistema (Non implementato)



2.3 Mapping hardware-software



2.3.1 Web Server

Il server utilizzato è Apache versione 9.

2.3.2 Interface layer

L'utente utilizza il sistema mediante un Browser installato all'interno del suo calcolatore (ad es. Opera, Firefox, Chrome).

2.3.3 Application Logic layer

Il sistema, e quindi le funzionalità, sono implementate in linguaggio HTML5 + java.

2.3.4 Storage layer

Rappresenta il collegamento con il server da parte del sistema e si occupa di tutte le richieste di accesso e modifiche sui dati permanenti presenti nel database.

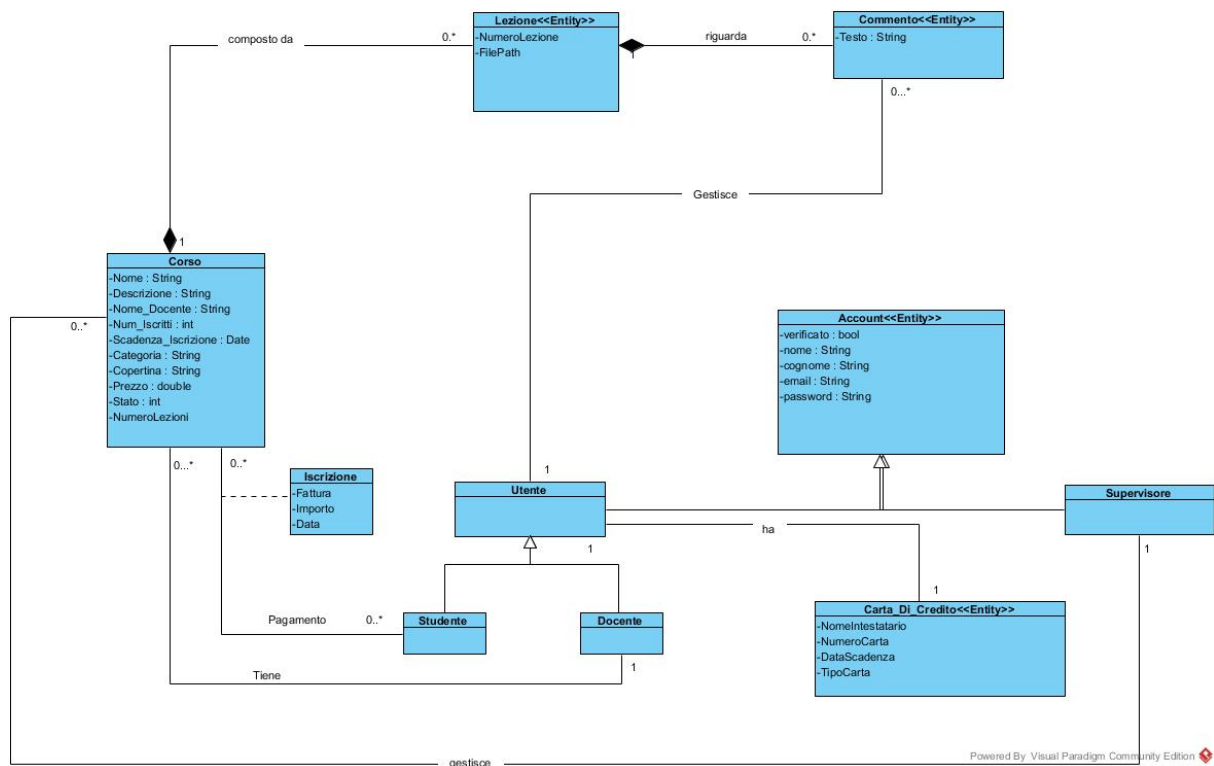
2.3.5 Database Server

Il DBMS usato è MySQL.

2.4 Gestione dati persistenti

All'interno di questa sezione verrà illustrato il procedimento di design sui dati persistenti partendo dal diagramma delle classi composto dagli entity per poi arrivare alla progettazione di un database relazionale.

Di seguito riportiamo il class diagram ripreso dal RAD di YouLearn:



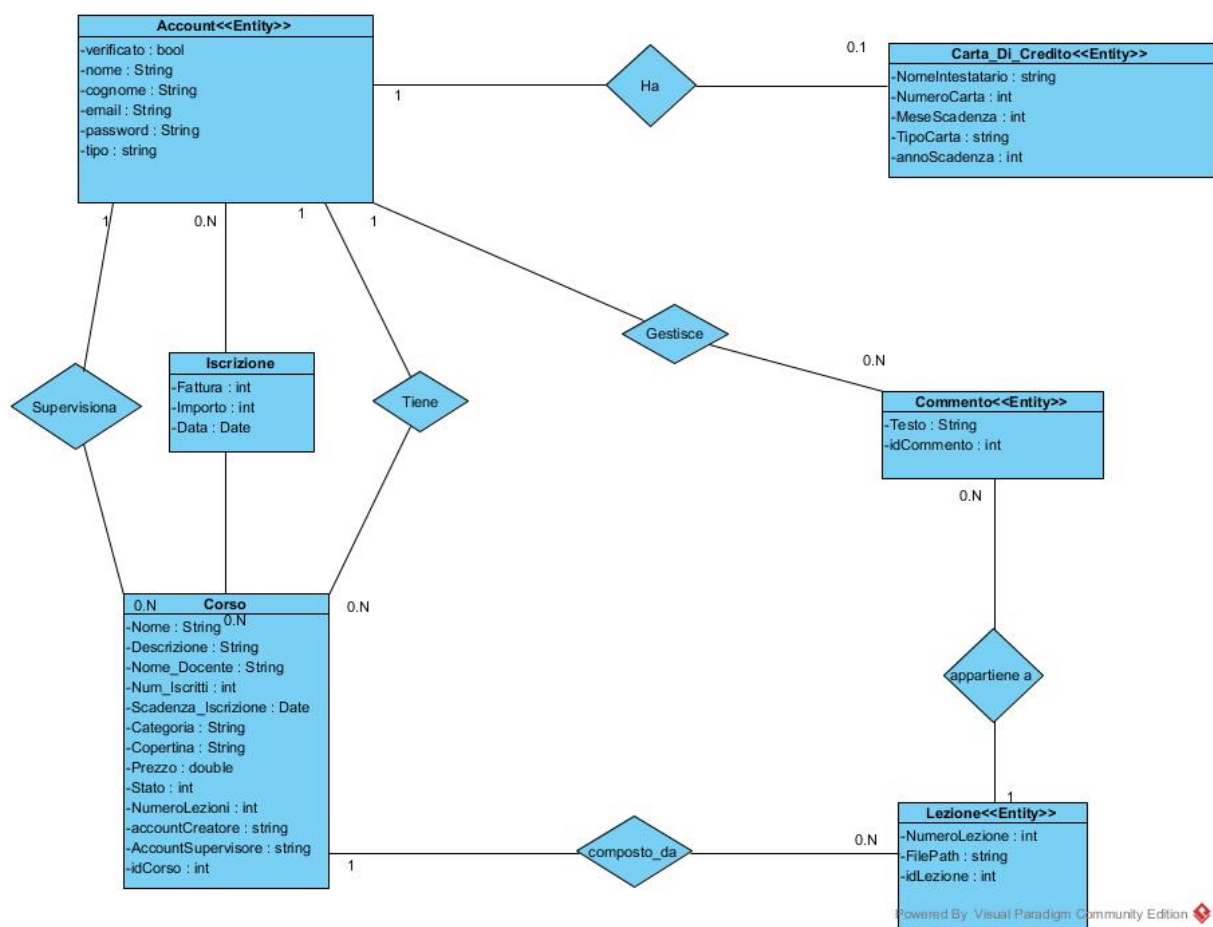
Sono state eseguite le seguenti modifiche per la stesura del diagramma ER:

- La classe Utente, che poteva essere un docente o uno studente, viene ora rappresentata da un'unica tabella "Utente" in quanto un Utente, nel sistema, è sia un Docente che uno Studente.
- La classe Account, che poteva essere un Utente o un Supervisore è ora rappresentata dalla tabella Account, il tipo di Account è specificato dal valore dell'attributo "Tipo".
- La classe Corso è rappresentata dalla tabella Corso: l'attributo "Stato" specifica lo stato in cui si trova il corso con un intero da 1 a 4:
 - 1- In creazione;
 - 2- In valutazione;
 - 3- Attivo;
 - 4- Disattivato.Attraverso l'associazione "Tiene" viene identificato un Utente che tiene il corso;

attraverso l'associazione "Paga" viene identificato un Utente che paga, e quindi frequenta, il corso; attraverso l'associazione "Supervisiona" si identifica il Supervisore che controlla il corso.

- La classe Commenti è rappresentata dalla tabella "Commento": attraverso l'associazione "Scrive" si identifica l'utente che ha scritto il commento; attraverso l'associazione "appartieneA" si identifica la Lezione a cui il commento appartiene.
- Le lezioni dei corsi sono rappresentate dalla tabella "Lezioni": l'associazione "compostoDa" rappresenta il corso a cui una lezione appartiene.
- La classe "Carta di Credito" è rappresentata dalla tabella "Carta di Credito": l'associazione "ha" rappresenta l'Utente che ha registrato quella carta di credito

Di seguito viene riportato il diagramma EER e le tabelle del database affiliate:



2.4.1 Account

NOME	TIPO	VINCOLI	KEY
Nome	VARCHAR(25)	NOT NULL	
Cognome	VARCHAR(25)	NOT NULL	
Password	VARCHAR(45)	NOT NULL	
Email	VARCHAR(255)	NOT NULL	PRIMARY KEY
Tipo	INTEGER(10)	NOT NULL	
Verificato	TINYINT(1)	NOT NULL	

2.4.2 Corso

NOME	TIPO	VINCOLI	KEY
idCorso	INT	NOT NULL, AUTO_INCREMENT	PRIMARY KEY
AccountCreatore	VARCHAR(255)	NOT NULL	FOREIGN KEY
AccountSupervisore	VARCHAR(255)	NOT NULL	FOREIGN KEY
Nome	VARCHAR(255)	NOT NULL	
Descrizione	VARCHAR(1048)	NOT NULL	
DataCreazione	DATE	NOT NULL	
DataFine	DATE	NOT NULL	
Copertina	VARCHAR (255)	NOT NULL	
Prezzo	VARCHAR(10)	NOT NULL	
Stato	INTEGER(10)	NOT NULL	
Categoria	VARCHAR(30)	NOT NULL	
NLezioni	INTEGER (10)	NOT NULL	
NIscritti	INTEGER(10)	DEFAULT 0	

2.4.3 Iscrizione

NOME	TIPO	VINCOLI	KEY
AccountMail	VARCHAR(255)	NOT NULL	PRIMARY KEY, FOREIGN KEY
CorsoIDCorso	INT	NOT NULL	PRIMARY KEY, FOREIGN KEY
DataPagamento	DATE	NOT NULL	
Importo	DOUBLE (10)	NOT NULL	
Fattura	INTEGER(10)	NOT NULL	UNIQUE

2.4.4 Carta di credito

NOME	TIPO	VINCOLI	KEY
AccountMail	VARCHAR(255)	NOT NULL	FOREIGN KEY
NumeroCarta	VARCHAR(16)	NOT NULL	PRIMARY KEY
MeseScadenza	VARCHAR (20)	NOT NULL	
AnnoScadenza	VARCHAR (20)	NOT NULL	
Tipo	INTEGER (5)	NOT NULL	
NomeIntestatario	VARCHAR(50)	NOT NULL	

2.4.5 Lezione

NOME	TIPO	VINCOLI	KEY
CorsoldCo	INTEGER (10)	NOT NULL	PRIMARY KEY FOREIGN KEY
Nome	VARCHAR(255)	NOT NULL	
Visualizzazioni	INTEGER(32)	NOT NULL	
NumeroLezione	INTEGER(10)	NOT NULL	
IdLezione	Integer(10)	NOT NULL	PRIMARY KEY

2.4.6 Commento

NOME	TIPO	VINCOLI	KEY
IdCommento	INT	NOT NULL, AUTO_INCREMENT	PRIMARY KEY
Testo	VARCHAR(255)	NOT NULL	
AccountMail	VARCHAR(255)	NOT NULL	FOREIGN KEY
lezioneldLezione	INTEGER (10)	NOT NULL	FOREIGN KEY

2.5 Controllo degli accessi e sicurezza

All' interno del sistema i vari attori hanno il permesso di eseguire operazioni diverse sui vari sottosistemi, sono escluse le operazioni che non verranno implementate in questa versione del software, quali sistema di gestione delle e-mail e dei pagamenti dei corsi. Per schematizzare meglio il controllo si è deciso di usare la matrice degli accessi riportata di seguito:

Attori - Sottosistemi	Studente	Docente	Supervisore
Utenti	<ul style="list-style-type: none"> ✕ Login ✕ Logout ✕ Visualizzazione info personali ✕ Modifica e-mail, password e carta di credito. 	<ul style="list-style-type: none"> ✕ Login ✕ Logout ✕ Visualizzazione info personali. ✕ Modifica e-mail, password e carta di credito 	<ul style="list-style-type: none"> ✕ Login ✕ Logout ✕ Visualizzare info personali ✕ Modifica e-mail o password.
Corsi	<ul style="list-style-type: none"> ✕ Visionare corsi ✕ Iscrivere ad un corso 	<ul style="list-style-type: none"> ✕ Creare un corso ✕ Modificare un corso ✕ Eliminare un corso 	<ul style="list-style-type: none"> ✕ Verificare un corso
Lezioni	<ul style="list-style-type: none"> ✕ Visualizzare una lezione ✕ Inserire un commento ✕ Eliminare un commento 	<ul style="list-style-type: none"> ✕ Inserire una lezione ✕ Eliminare una lezione ✕ Inserire un commento ✕ Eliminare un commento ✕ Ordinare le lezioni ✕ Modificare una lezione 	

Mail			
Pagamento			

2.6 Controllo Globale del software

Il flusso di controllo globale è la sequenza di azioni nel sistema. Il sistema ha un flusso guidato di eventi; le funzionalità richiedono un'interazione continua da parte dell'utente; per questo motivo, il controllo del flusso globale che utilizziamo è di tipo procedure-driven. Quindi non abbiamo una sequenza di operazioni prestabilite ma, è l'utente che sceglie l'operazione da eseguire

2.7 Condizione di boundary

[UC1] UC_STARTUP

ID:	UC_STARTUP					
Caso d'uso:	Startup del server					
Istanze di Attori partecipanti:	Davide: Amministratore					
Condizione di entrata:	L'amministratore accede al sistema					
Condizione di uscita:	Il server è attivo e i relativi servizi sono disponibili					
Flusso di eventi:	<table><tr><td>Amministratore</td><td>Sistema</td></tr><tr><td>Davide avvia il server del sistema cliccando su "Avvia"</td><td><p>Il sistema si avvia e attiva i servizi in remoto andando a renderli disponibili per le richieste.</p><p>Il sistema notifica all'utente che l'operazione di startup del server è avvenuta con successo</p></td></tr></table>		Amministratore	Sistema	Davide avvia il server del sistema cliccando su "Avvia"	<p>Il sistema si avvia e attiva i servizi in remoto andando a renderli disponibili per le richieste.</p> <p>Il sistema notifica all'utente che l'operazione di startup del server è avvenuta con successo</p>
Amministratore	Sistema					
Davide avvia il server del sistema cliccando su "Avvia"	<p>Il sistema si avvia e attiva i servizi in remoto andando a renderli disponibili per le richieste.</p> <p>Il sistema notifica all'utente che l'operazione di startup del server è avvenuta con successo</p>					
Eccezioni:	Durante la fase di avvio del server, se si verifica un errore, l'amministratore riceverà una notifica di errore.					

[UC2] UC_SHUTDOWN

ID:	UC_SHUTDOWN				
Caso d'uso:	Shutdown del server				
Istanze di Attori partecipanti:	Davide: Amministratore				
Condizione di entrata:	L'amministratore accede al sistema				
Condizione di uscita:	Il server è attivo e i relativi servizi sono disponibili				
Flusso di eventi:	<table><thead><tr><th>Amministratore</th><th>Sistema</th></tr></thead><tbody><tr><td>Davide spegne il sistema cliccando su "Spegni".</td><td>Il sistema effettua una scansione per verificare che non ci siano richieste in attesa, se vi sono, porta a termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.</td></tr></tbody></table>	Amministratore	Sistema	Davide spegne il sistema cliccando su "Spegni".	Il sistema effettua una scansione per verificare che non ci siano richieste in attesa, se vi sono, porta a termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.
Amministratore	Sistema				
Davide spegne il sistema cliccando su "Spegni".	Il sistema effettua una scansione per verificare che non ci siano richieste in attesa, se vi sono, porta a termine tali richieste e, quando non ne ha più in sospeso, notifica all'amministratore l'inizio della procedura di spegnimento che sta eseguendo per poi terminare.				
Eccezioni:	Durante la fase di spegnimento, se si verifica un qualsiasi errore, questo viene notificato all'amministratore.				

2.7.1 Fallimento

I casi di fallimento del sistema possono essere di diverse tipologie:

1. Guasti dovuti al sovraccarico e successivo fallimento del database. In questo caso, come provvedimento, si ha quello del salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.
2. Interruzione inaspettata dell'alimentazione, non sono previsti metodi di ripristino dello stato del sistema prima dello spegnimento.
3. Software che causa la chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. In questo caso non sono previste politiche di correzione; l'unico processo eseguibile, sarà la chiusura del sistema e il suo riavvio.
4. **Errore critico nell'hardware. In questo caso, momentaneamente, non sono previste soluzioni.**

3. Servizi dei sottosistemi

In questa sezione è possibile visionare i vari sottosistemi con relativi servizi offerti, richiesti e descrizione informative affiliate, saranno presenti anche i sottosistemi che non verranno implementati per definire una visione completa del sistema desiderato.

3.1 Gestione corsi

Sottosistema	Autenticazione
Descrizione	Sottosistema che gestisce tutte le operazioni riguardante i corsi del sistema.
Servizi offerti	Descrizione
Crea corso	Questa funzionalità permette di creare un corso ex-novo
Modifica Corso	Questa funzionalità permette di modificare un corso in via di creazione
Convalida Corso	Questa funzionalità permette ad un Supervisore di confermare o rifiutare un corso appena creato
Conferma Corso	Questa funzionalità permette ad un Utente di confermare un corso in fase di creazione e passarlo alla fase di supervisionamento.
Visualizza Corso	Questa funzionalità permette di visionare un corso
Elimina corso	Questa funzionalità permette di eliminare un corso in via di creazione
Iscrizione corso	Questa funzionalità permette di iscriversi ad un corso
Servizi richiesti	Descrizione
Pagamento	Questa funzionalità è richiesta per il servizio

	"Iscrizione Corso"
Mail	Questa funzionalità è richiesta per il servizio "Verifica Corso"

3.2 Gestione Lezioni:

Sottosistema		Autenticazione	
Descrizione		Sottosistema che gestisce tutte le operazioni riguardanti le lezioni	
Servizi offerti		Descrizione	
Elimina lezione		Questa funzionalità permette di eliminare un a lezione da corso in via di creazione	
Inserisci lezione		Questa funzionalità permette di inserire una lezione ad un corso in creazione	
Visualizza lezione		Questa funzionalità permette di visualizzare una lezione	
Modifica lezione		Questa funzionalità permette di modificare una lezione	
Modifica ordine lezione		Questa funzionalità permette di variare il numero della lezione per un gruppo di lezioni di un corso che si trova in fase di completamento.	
Elimina commento		Questa funzionalità permette di eliminare un commento scritto da sé stessi o uno qualsiasi relativo a lezioni di corsi propri	
Inserisci commento		Questa funzionalità permette di inserire un commento ad una lezione di un proprio corso o di un corso che si segue	
Servizi richiesti		Descrizione	

3.3 Gestione Pagamento:

Sottosistema	Autenticazione
Descrizione	Sottosistema che gestisce tutte le operazioni riguardanti i pagamenti
Servizi offerti	Descrizione
InviaPagamento	Questa funzionalità permette di effettuare il passaggio di denaro per l'acquisto di un corso
Servizi richiesti	Descrizione

3.4 Gestione mail:

Sottosistema	Autenticazione
Descrizione	Sottosistema che gestisce tutte le operazioni riguardante le informazioni dell'account.
Servizi offerti	Descrizione
MailConfermaAccount	Questa funzionalità permette di inviare una mail per la conferma di un account Utente
MailVerificaCorso	Questa funzionalità permette di inviare una mail notificando l'avvenuta verifica di un proprio corso ad un utente
MailRecuperoPassword	Questa funzionalità permette di modificare la password di un Account per cui non si ricorda la password originale
Servizi richiesti	Descrizione

3.5 Gestione Utente:

Sottosistema		Autenticazione	
Descrizione		Sottosistema che gestisce tutte le operazioni riguardante gli utenti del sito.	
Servizi offerti		Descrizione	
Login		Questa funzionalità permette di effettuare l'accesso al sistema inserendo le proprie credenziali.	
Logout		Questa funzionalità permette di uscire dal sistema.	
Modifica email		Questa funzionalità permette di modificare la mail inserita alla registrazione	
Modifica password		Questa funzionalità permette di modificare la password inserita alla registrazione	
Modifica carta		Questa funzionalità permette di modificare la carta inserita alla registrazione	
Recupero Password		Questa funzionalità permette di recuperare la password di un account	
Registrazione Utente		Questa funzionalità permette di registrarsi al sistema e quindi inserire un nuovo utente	
Servizi richiesti		Descrizione	
Mail		Questa funzionalità è richiesta per il servizio "Recupero Password"	

4. Glossario

Client: componente che accede ai servizi o alle risorse di un'altra componente, detta server.

Deployment Diagram: Schema che descrive la struttura dinamica del sistema

DBMS: programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.

Login: Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.

Layer: È un insieme di classi con funzionalità simile (tipicamente raggruppati in un unico package).

Logout: Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password (vedi login).

MySQL: Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.

Server: Programma di gestione di un servizio che invia informazioni in un particolare formato ricevuto e interpretato da un programma Client dal lato ricevente.

Shutdown: fase di terminazione del sistema.

Start-up: processo di accensione e di avvio di un computer, di un dispositivo o di un sistema.

MVC: Pattern architetturale che vede la divisione in tre parti cooperanti: Model, View, Control.