

UNIVERSITÀ DEGLI STUDI DI SALERNO



INGEGNERIA DEL SOFTWARE

System Design Document (SDD)



Coordinatori del Progetto
Prof.ssa Filomena Ferrucci
Liliana Annunziata
Raffaele Costantino

Partecipanti
Alessandro Kevin Barletta
Edoardo Carpentiero
Stefano Cirillo
Andrea De Maio
Gianmarco Mucciariello
Antonio Pizza
Alberto Sergio
Gianmaria Trezza

Revision History

Data	Versione	Descrizione	Autori
01/12/2015	0.1	Introduzione e glossario	Andrea De Maio Alberto Sergio
01/12/2015	0.2	Architettura del sistema corrente Architettura del sistema proposto	Stefano Cirillo Gianmaria Trezza
02/12/2015	0.3	Gestione dati persistente Controllo degli accessi Condizioni Boundary Gestione del controllo globale	Stefano Cirillo Gianmaria Trezza
03/12/2015	0.4	Architettura del sistema proposto	Alberto Sergio Gianmaria Trezza
04/12/2015	0.5	Decomposizione in sottosistemi	Alessandro Kevin Barletta Gianmarco Mucciariello
04/12/2015	0.6	Servizi dei sottosistemi	Alessandro Kevin Barletta Gianmarco Mucciariello
05/12/2015	1.0	Revisione del SDD	Gianmarco Mucciariello
07/12/2015	1.1	Aggiunta sottosistema gestione autenticazione	Team Members
08/12/2015	1.2	Modifica sottosistema gestione account	Team Members
22/12/2015	2.0	Revisione del SDD	Alberto Sergio Andrea De Maio Gianmaria Trezza
22/01/2016	2.1	Controllo SDD	Alberto Sergio
27/01/2016	3.0	Revisione SDD	Gianmarco Mucciariello

Sommario

1 Introduzione

1.1 Scopo del sistema

1.2 Design Goal

1.2.1. Trade-off

1.2.1.1. TO_0 Spazio vs efficienza

1.2.1.2. TO_1 Tempo di rilascio vs qualità

1.2.1.3. TO_2 Tempo di rilascio vs funzionalità

1.2.1.4. TO_3 Efficienza vs portabilità

1.3 Riferimenti

1.4 Overview

Nei successivi punti del System Design Document verranno descritti:

2 Architettura del sistema corrente

3 Architettura del sistema proposto

3.1 Overview

3.2 Decomposizione in sottosistemi

3.2.1 Sottosistema generale

3.2.2 SubD_0 per Sottosistema gestione autenticazione

SottosistemaGestioneAutenticazione include tutte le componenti del sistema che offrono le funzionalità per la gestione dell'autenticazione.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

3.2.3 SubD_1 per Sottosistema gestione docenti

3.2.4 SubD_2 per Sottosistema gestione insegnamenti

3.2.5 SubD_3 per Sottosistema gestione ordinamento

3.2.6 SubD_4 per Sottosistema gestione regolamento

3.2.7 SubD_5 per Sottosistema gestione programmazione didattica

3.2.8 SubD_6 per Sottosistema gestione account

3.3 Hardware/software Mapping

3.4 Gestione dati persistenti

3.5 Controllo degli accessi e sicurezza

3.6 Gestione del controllo globale

3.7 Condizioni Boundary

4 Servizi dei sottosistemi

4.1 SS_0 Sottosistema per RF_0 Gestione Autenticazione

4.2 SS_1 Sottosistema per RF_1 Gestione Docenti

4.3 SS_2 Sottosistema per RF_2 Gestione Insegnamenti

4.4 SS_3 Sottosistema per RF_3 Gestione Ordinamento

4.5 SS_4 Sottosistema per RF_4 Gestione Regolamento

4.7 SS_6 Sottosistema per RF_6 Gestione Account

5 Glossario

1 Introduzione

1.1 Scopo del sistema

L'obiettivo del software Pr. D. è quello di mettere a disposizione un sistema che faciliti le normali operazioni svolte dal Presidente del Consiglio Didattico di Informatica per gestire la Programmazione Didattica.

Pr.D, inoltre, permette anche l'accesso a tutti i docenti che sono dotati di un proprio account. Questi grazie all'accesso al sistema possono effettuare tutte le operazioni di visualizzazione fornite da esso, tra cui anche la visualizzazione della Programmazione Didattica da approvare. Inoltre possono cambiare il loro stato(attivo/inattivo), in modo tale che il Presidente al momento della pianificazione della Programmazione Didattica può controllare le disponibilità di tutti i Docenti.

Infine, Pr.D, fornisce informazioni riguardanti la programmazione didattica all'utente.

La progettazione del sistema pone come obiettivo quello di garantire le caratteristiche di usabilità, persistenza dei dati, ed affidabilità del sistema, oltre ai requisiti funzionali e non funzionali individuati nelle fasi precedenti.

1.2 Design Goal

Gli obiettivi del design vengono sviluppati basandosi sui requisiti di qualità identificati nel sistema, i quali garantiscono che il sistema sia capace di offrire prestazioni adeguate allo scopo per cui è stato progettato.

Di seguito vengono riportati gli obiettivi che sono perseguiti nella fase di implementazione del sistema Pr. D.

DG_0 Criteri di Performance	
DG_0.1 Tempo di risposta	E' stato predisposto che operazioni standard quali l'avvio del sistema e la risposta al click del mouse, dovranno essere quanto più brevi possibili (max 2 secondi) per consentire un uso in tempo reale. Per le operazioni più impegnative quali l'accesso ai dati memorizzati nel database potranno essere richiesti al più 3 secondi ulteriori. Questo livello di prestazione deve comunque essere legato al sistema hardware sul quale lo stesso sarà installato ed utilizzato.
DG_0.2 Troughput	Pr.D. è stato concepito per permettere l'accesso in maniera rapida, con la possibilità di farlo anche da più terminali. Si prevede che il sistema riesca a far accedere ai servizi offerti da Pr. D. un massimo di 5.000 utenti, in media 50 utenti.
DG_0.3 Memoria	Le richieste di memoria da parte del programma sono molto limitate. La memoria di massa richiesta varia in funzione della mole di dati da trattare; la memoria centrale consigliata è, invece, pari a quella consigliata per l'utilizzo del sistema operativo.

DG_1 Criteri di Affidabilità	
DG_1.1 Fault Tollerance	In presenza di errori, il sistema non permette di continuare ad operare. Verranno mostrati avvisi con la descrizione dell'errore.
DG_1.2 Robustezza	Il sistema è in grado di filtrare gli input, poiché prima di essere inviati ai vari moduli dell'applicazione, sarà verificata la loro correttezza tramite opportune routine.
DG_1.3 Sicurezza	Il sistema è realmente sicuro visto che il presidente, per effettuare l'accesso, deve inserire la combinazione username/password in suo possesso.
DG_1.4 Affidabilità	Il sistema deve essere in grado di effettuare le operazioni nella loro completezza, lasciando così i valori dei dati nella situazione precedente ad un crash.

DG_2 Criteri di Mantenimento	
DG_2.1 Estendibilità	Deve essere possibile aggiungere, in un secondo momento, altre funzionalità in Pr.D, senza dover stravolgere il design del sistema.
DG_2.2 Modificabilità	Il sistema, in caso di nuove esigenze, darà la possibilità di apportare modifiche o miglioramenti alle proprie funzionalità.
DG_2.3 Adattabilità	Il sistema è usabile solo nello specifico dominio applicativo per il quale è stato progettato.
DG_2.4 Portabilità	Il prodotto software è realizzato in HTML, CSS, PHP, JavaScript e AJAX, il che lo rende particolarmente trasportabile su sistemi operativi diversi da quello di sviluppo.
DG_2.5 Leggibilità	Il codice è scritto e formattato per garantirne la comprensione anche a persone esterne al team, ed include commenti per chiarire parti di codice particolarmente complesse.
DG_2.6 Tracciabilità dei requisiti	E' importante che il codice rispetti la struttura progettuale del sistema specificata durante la fase di analisi dei requisiti per riconoscere il prima possibile, in caso serva, il modulo da prendere in considerazione.

DG_3 Utente Finale	
DG_3.1 Utilità	Il sistema risponderà in modo esaustivo a tutte quelle che sono le necessità dell'utente.
DG_3.2 Usabilità	Il sistema sarà progettato per essere quanto più usabile possibile, in modo da ridurre lo sforzo dell'utente per intraprendere un determinato task. Il sistema deve risultare semplice da utilizzare anche ad utenti poco esperti.

1.2.1. Trade-off

1.2.1.1. TO_0 Spazio vs efficienza

A livello implementativo, per garantire una maggiore efficienza, è stato utilizzato AJAX in collaborazione con gli altri linguaggi di programmazione, in modo tale da non dover ricaricare ogni volta le pagine.

1.2.1.2. TO_1 Tempo di rilascio vs qualità

Nonostante la scelta di mantenere una qualità media abbastanza alta, si è stabilito di dare maggiore priorità ai tempi di consegna del sistema. Nonostante questa decisione, il software sarà privo di bug per le funzionalità implementate.

1.2.1.3. TO_2 Tempo di rilascio vs funzionalità

Nel caso di si presentino problemi durante l'implementazione del software, o vengano rilevati errori durante la fase di testing, il software potrebbe essere rilasciato con meno funzionalità di quelle previste. Tali funzionalità saranno aggiunte dopo la data di rilascio.

1.2.1.4. TO_3 Efficienza vs portabilità

Per quanto riguarda il lato client c'è una buona portabilità, in quanto l'applicativo software è basato sul web browser. Mentre per il lato server, se sarà necessario, la portabilità del software è garantito in quanto il sistema è realizzato con tecnologie per lo sviluppo per il web.

1.3 Riferimenti

Documento RAD del Sistema Software.

1.4 Overview

Nei successivi punti del System Design Document verranno descritti:

- **Architettura Software proposta:** documenta il modello di System Design del nuovo sistema, in cui vengono descritti:
 1. **Decomposizione in sottosistemi:** descrive la suddivisione del sistema in vari sottosistemi.
 2. **Hardware / Software Mapping:** in cui vengono prese decisioni per quanto riguarda le piattaforme hardware e software su cui il sistema verrà installato.
 3. **Gestione dati persistenti:** in cui vengono identificati gli oggetti persistenti.
 4. **Controllo degli accessi e sicurezza:** documenta il modello utente in riferimento all'accesso al sistema e problemi di sicurezza.
 5. **Gestione del controllo globale:** documenta la gestione e quindi il funzionamento del software implementato.
 6. **Condizioni Boundary:** in cui sono descritte le condizioni limite del sistema (Errori).
 7. **Servizi dei Sottosistemi:** descrive i servizi forniti da ciascun sottosistema cioè le operazioni svolte.

2 Architettura del sistema corrente

L'architettura proposta nel seguito non andrà a rimpiazzare nessuna struttura pre-esistente. Infatti la progettazione e lo sviluppo di essa segue, essenzialmente, i criteri della Greenfield Engineering.

3 Architettura del sistema proposto

3.1 Overview

L'architettura del sistema **Pr.D Programmazione Didattica** è basata su un'architettura **"Client/Server"**.

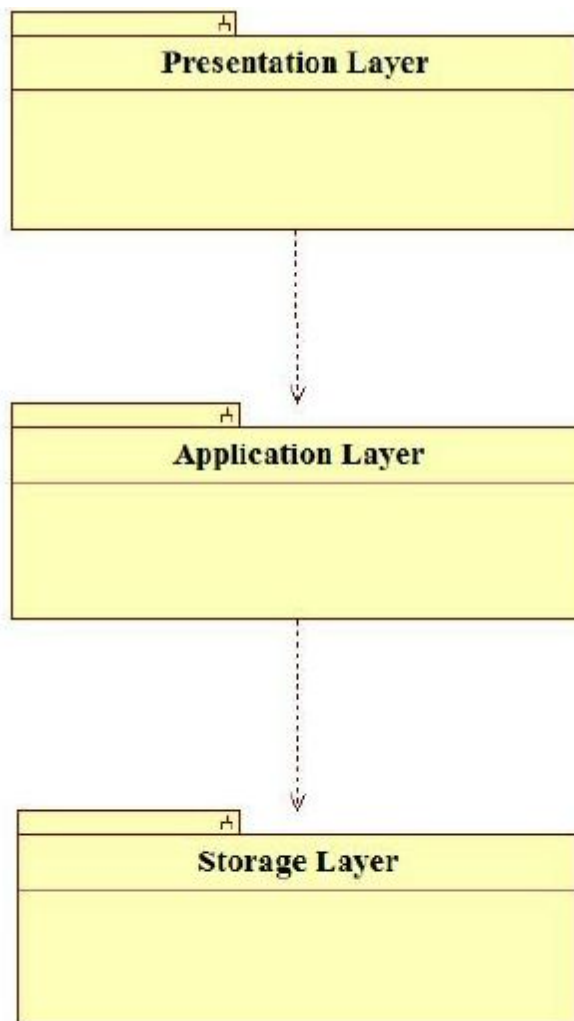
La scelta di questo tipo di architettura è indispensabile per l'uso della rete da parte del sistema. Infatti il sistema è progettato per utilizzare la rete Internet per collegarsi ad un server-data.

L'utilizzo della rete internet è impiegato sia dagli Utenti i quali la utilizzano per accedere ad informazioni generali sia dal Presidente il quale la utilizza per compiere azioni.

L'intera architettura **"Client/Server"** del sistema **Programmazione Didattica** è di tipo **"Three-Tier"**.

Prevediamo quindi una scomposizione del sistema in sottosistemi. Ogni sottosistema è suddiviso in tre strati:

- **Presentation Layer:** si occupa di presentare i risultati e raccogliere input degli utenti, questo include tutta la gestione delle interfacce utente.
- **Application Layer:** si occupa di fornire al software tutte le specifiche funzionalità per cui esso è stato progettato. Include la gestione degli oggetti e delle classi con relativi metodi e controlli.
- **Storage Layer:** si occupa della gestione del database del sistema e quindi include la memorizzazione e il recupero di oggetti persistenti.



L'immagine mostra gli strati di un sottosistema

Il sistema è basato sull' utilizzo di un database il quale conterrà al suo interno tutti i dati del sistema riguardanti la programmazione didattica, i regolamenti, i docenti e gli insegnamenti.

L' architettura “**Three-Tier**” è stata scelta perchè soddisfa tutti gli obiettivi che il sistema **Pr.D** si propone di raggiungere

Criteri:

1)Criteri di Performance: Le risorse vengono condivise, un server dedicato (database-server) si occupa della memorizzazione dei dati quindi abbiamo un'ottimizzazione dello spazio usato. La velocità è garantita dalla suddivisione dei compiti ovvero dalla decomposizione in sottosistemi infatti, mentre la macchina client esegue l'interfaccia grafica vengono formulate delle richieste al server-dataBase che si occupa di fornire i servizi e le funzionalità del sistema e infine il server-dataBase soddisfa le richieste restituendo i dati a cui si è interessati.

2) Criteri di Affidabilità: L'affidabilità è garantita dai backup continui che vengono effettuati dal server-data.

3) Criteri di Costo: Il nostro sistema è un sistema **Greenfield Engineering** quindi non esistono costi di conversione ad un sistema precedentemente sviluppato per poter garantire la compatibilità. I costi ridotti sono garantiti dall'uso della rete Internet da parte del sistema anziché di una rete dedicata. I costi di manutenzione sono pressoché inesistenti a meno che non si decida di effettuare modifiche radicali allo schema dei dati, anche se ciò accadesse, grazie alla tipologia di architettura scelta si riesce ad individuare in maniere tempestiva il modulo su cui agire e minimizzare i costi di un aggiornamento.

4) Criteri di Mantenimento: con l'architettura "Three-Tier" il nostro sistema viene diviso in sottosistemi. Ogni sottosistema è suddiviso in tre layer(livelli):

- PresentationLayer
- ApplicationLayer
- StorageLayer

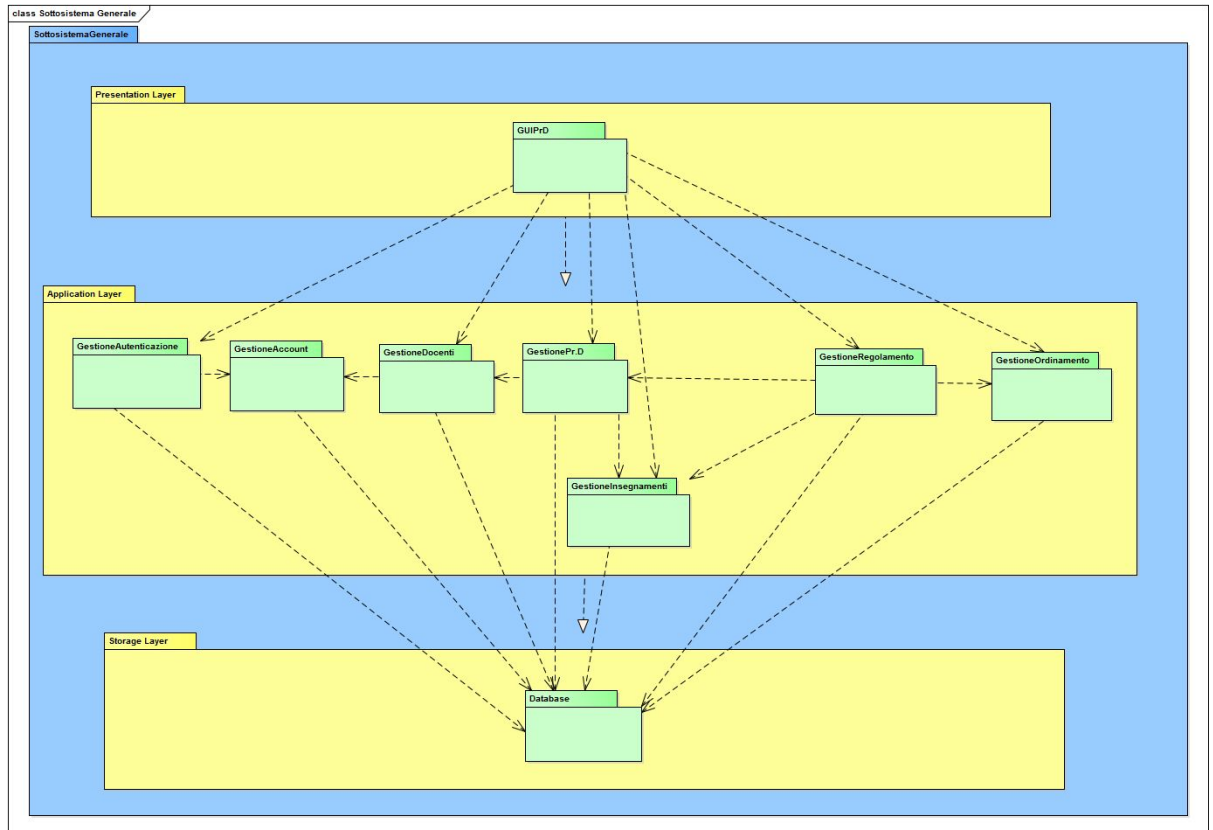
L'ApplicationLayer è realizzato da moduli che hanno uno specifico compito. Questa organizzazione garantisce la modularità del sistema di conseguenza l'estendibilità e la modificabilità, infatti risulterebbe semplice aggiungere nuove funzionalità al programma inserendo nuovi moduli nello strato di Application Layer di un sottosistema.

Il sistema garantisce anche un'ottima portabilità infatti l'interfaccia grafica è stata creata per poter essere eseguita da qualunque tipo di macchina e con qualunque sistema operativo. I Client, con macchine che abbiano caratteristiche completamente diverse tra loro, possono avere accesso ai dati effettuando richieste al server-data e ottenendo gli stessi risultati.

5) Criteri di End User: L'usabilità del sistema è garantita da un'interfaccia grafica molto chiara e dettagliata. I pulsanti sono ben visibili e con nomi che fanno comprendere facilmente il loro compito, l'interfaccia grafica è personalizzabile con colori a piacere, in ogni finestra sono presenti pulsanti che permettono di tornare indietro e di annullare l'operazione. Tutte queste caratteristiche dell'interfaccia infondono tranquillità e sicurezza anche all'utente meno esperto, sia che si tratti di un Utente esterno che usa l'interfaccia Web e sia che si tratti del Presidente che utilizza l'interfaccia grafica di gestione della Programmazione Didattica.

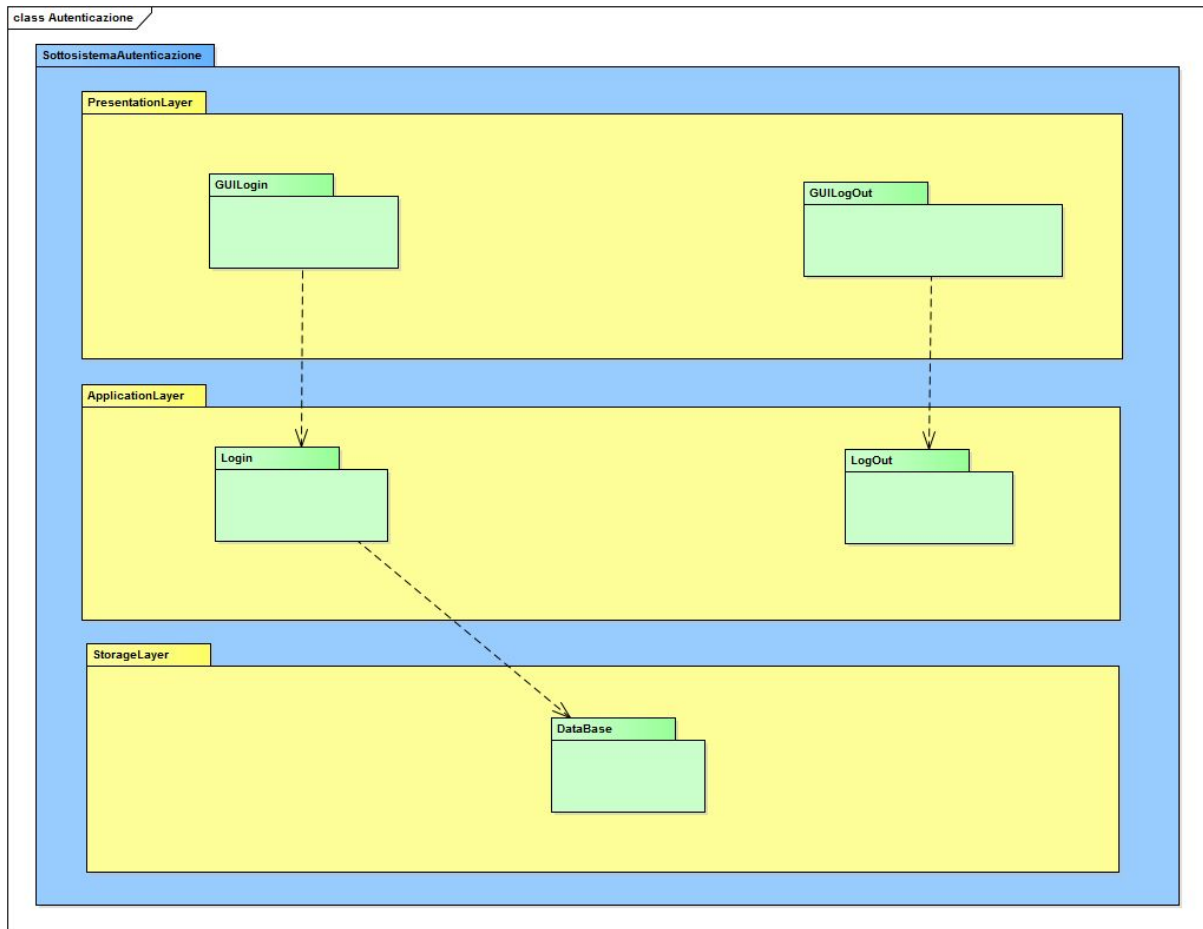
3.2 Decomposizione in sottosistemi

3.2.1 Sottosistema generale



All'interno del **PresentationLayer** si trova l'interfaccia del sistema, questa interfaccia consente di utilizzare le funzioni del sistema che si trovano all'interno dell'**ApplicationLayer**, ovvero le varie gestioni. Infine tutti i dati vengono memorizzati all'interno di un **Database**.

3.2.2 SubD_0 per Sottosistema gestione autenticazione



SottosistemaGestioneAutenticazione include tutte le componenti del sistema che offrono le funzionalità per la gestione dell'autenticazione.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUILogin**: è l'interfaccia costituita da un modulo contenente due caselle di testo per l'inserimento di username e password ed un pulsante per effettuare il login.
- **GUILogout**: è l'interfaccia costituita da un bottone per effettuare il logout.

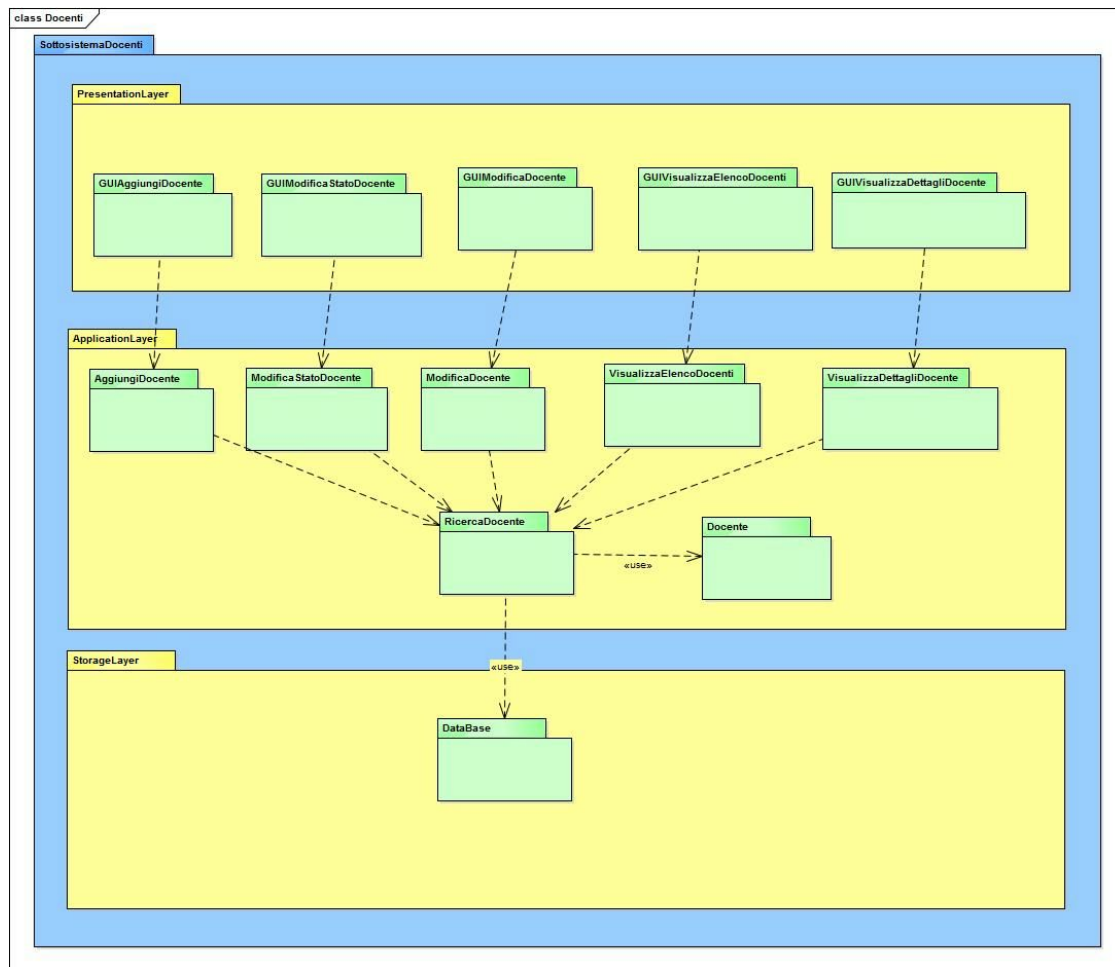
ApplicationLayer:

- **Login**: Si occupa di effettuare l'operazione di accesso al sistema, interagendo con il database per verificare la correttezza dell'username e della password.
- **Logout**: Si occupa di effettuare l'operazione di uscita dal sistema.

StorageLayer:

- **Database**: è il database contenente tutti gli utenti registrati al sistema..

3.2.3 SubD_1 per Sottosistema gestione docenti



SottosistemaGestioneDocenti include tutte le componenti del sistema che offrono le funzionalità per la gestione dei docenti.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIAggiungiDocente**: è l'interfaccia costituita da un modulo contenente una casella di testo per ogni informazione necessaria alla creazione di un nuovo docente, ed un bottone per l'aggiunta.
- **GUIVisualizzaDocente**: è l'interfaccia costituita da due caselle di testo per l'inserimento dei parametri di ricerca ed un bottone per la visualizzazione.
- **GUIVisualizzaElencoDocenti**: è l'interfaccia costituita da un elenco contenente tutti i docenti appartenenti alla facoltà di Informatica.
- **GUIModificaDocente**: è l'interfaccia costituita da un modulo contenente una casella di testo per ogni informazione modificabile di un docente, ed un bottone per la modifica
- **GUIModificaStatoDocente**: è l'interfaccia costituita da un modulo contenente due caselle di testo per la ricerca di un docente, ed un bottone per effettuare la modifica di stato.

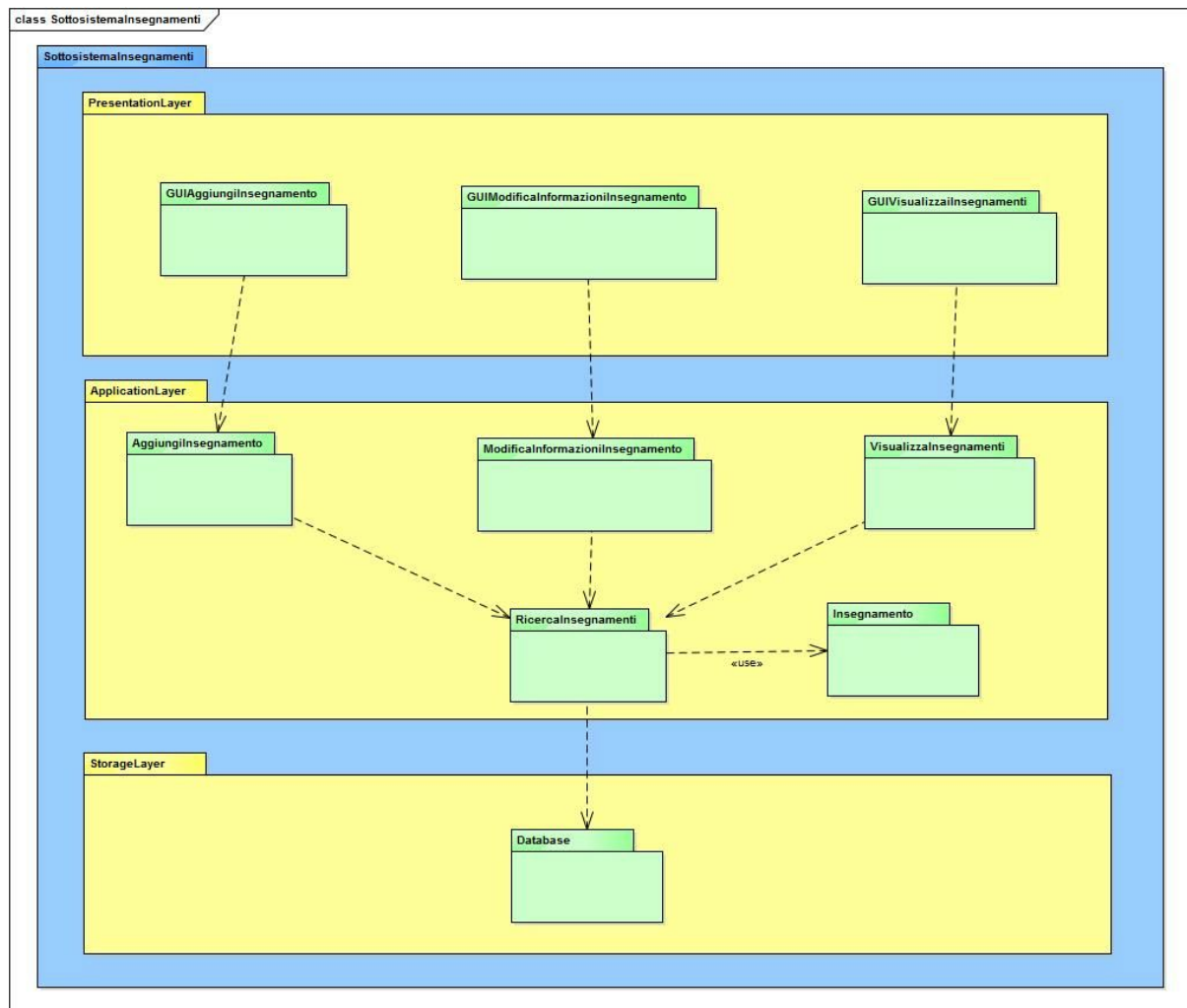
ApplicationLayer:

- **AggiungiDocente:** Si occupa di effettuare l'operazione di inserimento di un docente, interagendo con il database per verificare la correttezza dei dati.
- **VisualizzaDettagliDocente:** Si occupa di mostrare all'utente i docenti da visualizzare, cercandoli nel database secondo i dati inseriti dall'utente.
- **VisualizzaElencoDocenti:** Si occupa di mostrare all'utente l'elenco dei docenti presenti nel database.
- **ModificaDocente:** Si occupa di effettuare il cambiamento dei dati di un dato docente e di comunicarli al database.
- **ModificaStatoDocente:** Si occupa di effettuare il cambiamento di stato di un dato docente e di comunicarlo al database.
- **Ricerca docente:** Si occupa di effettuare la ricerca di un docente all'interno del database.
- **Docente:** si occupa della gestione di tutte le operazioni riguardanti i docenti.

StorageLayer:

- **Database:** è il database contenente tutti i docenti presenti.

3.2.4 SubD_2 per Sottosistema gestione insegnamenti



SottosistemaGestioneInsegnamenti, include tutte le componenti che permettono al Presidente di eseguire le operazioni di aggiunta, modifica e visualizzazione, mentre l'unica operazione consentita all'utente è quella di visualizzazione.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIVisualizzaInsegnamenti**: è un'interfaccia che permette di visualizzare una data tipologia di insegnamenti. Può essere utilizzata sia dall'utente che dal presidente.
- **GUIAggiungiInsegnamento**: è un'interfaccia che consente di aggiungere un nuovo insegnamento all'interno del sistema.
- **GUIModificaInformazioniInsegnamento**: è un'interfaccia che consente di modificare un dato insegnamento.

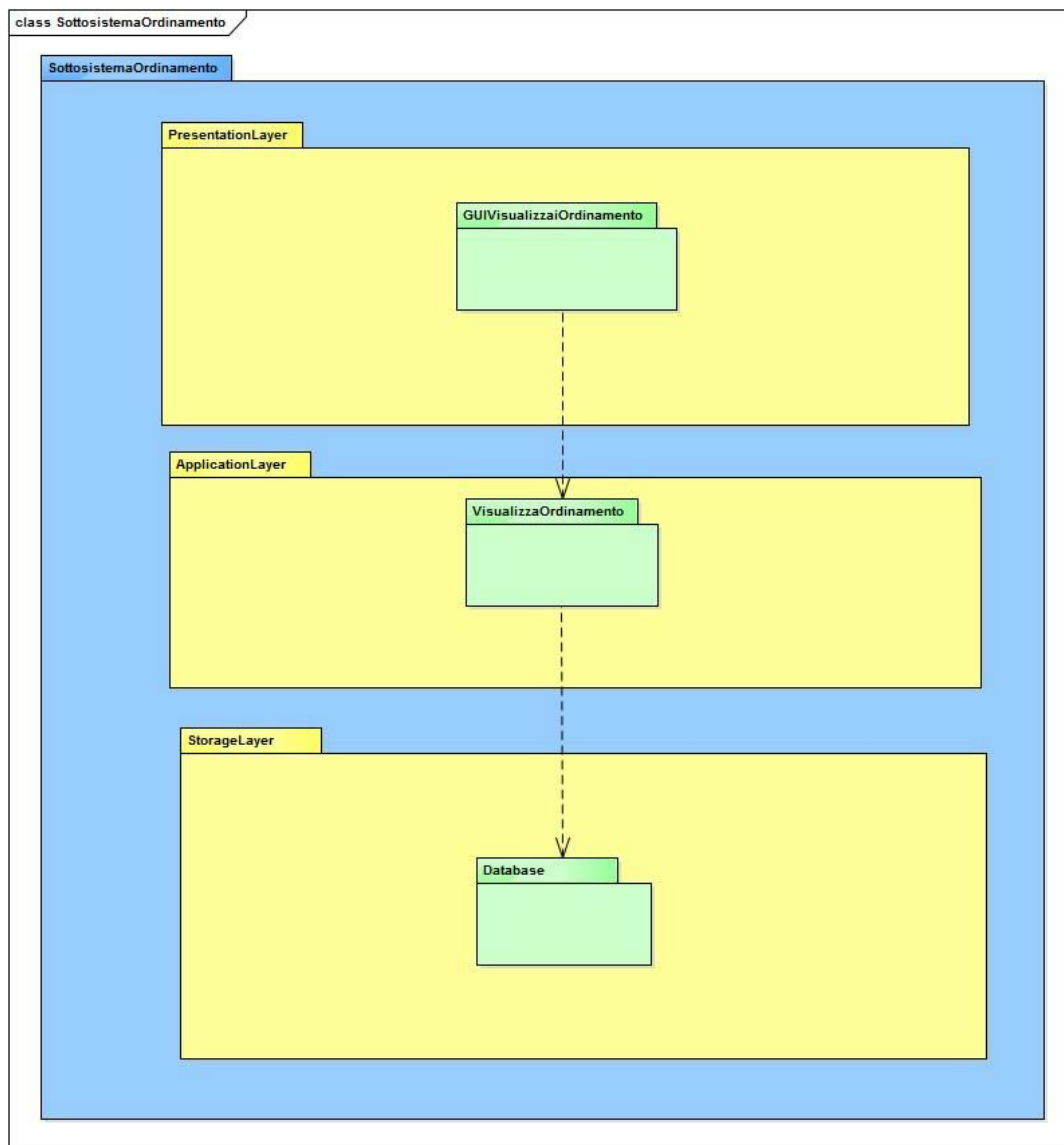
ApplicationLayer:

- **RicercaInsegnamenti:** permette di eseguire la ricerca di una specifica tipologia di insegnamento.
- **VisualizzaInsegnamenti:** visualizza gli insegnamenti per cui è stata eseguita la ricerca.
- **AggiungiInsegnamento:** aggiunge un nuovo insegnamento all'interno del sistema e utilizza "**RicercaInsegnamenti**" per verificare che l'insegnamento che si vuole inserire non si trovi già all'interno del sistema.
- **ModificaInformazioniInsegnamento:** permette la modifica delle informazioni relative ad un insegnamento.
- **Insegnamento:** si occupa della gestione di tutte le operazioni riguardanti gli insegnamenti.

StorageLayer:

- **Database:** è il database in cui vengono memorizzate tutte le informazioni relative agli insegnamenti.

3.2.5 SubD_3 per Sottosistema gestione ordinamento



SottosistemaGestioneOrdinamento, include l'operazione che permette al presidente e all'utente di visualizzare l'ordinamento.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIVisualizzaOrdinamento**: permette di visualizzare l'ordinamento.

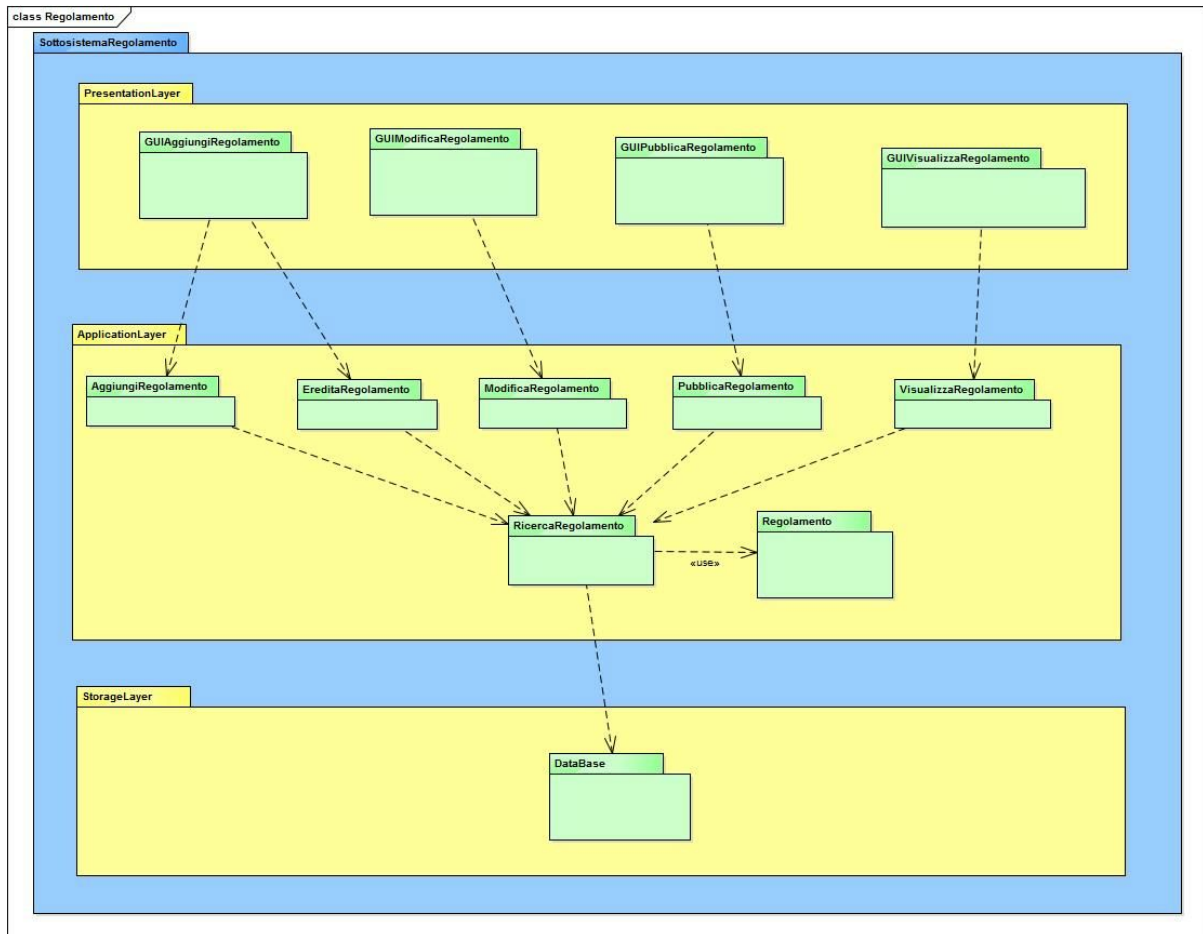
ApplicationLayer:

- **VisualizzaOrdinamento**: La funzione che visualizza l'ordinamento richiesto

StorageLayer:

- **Database**: è il database in cui vengono memorizzate tutte le informazioni relative all'ordinamento.

3.2.6 SubD_4 per Sottosistema gestione regolamento



SottosistemaGestioneRegolamento Include tutte le componenti del sistema che offrono le funzionalità per la gestione del regolamento.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIAggiungiRegolamento:** è l'interfaccia costituita da un modulo contenente una casella di testo per ogni informazione necessaria alla creazione di un nuovo regolamento, ed un bottone per l'aggiunta.
- **GUIVisualizzaRegolamento:** è l'interfaccia costituita da due caselle di testo per l'inserimento dei parametri di ricerca ed un bottone per la visualizzazione.
- **GUIModificaRegolamento:** è l'interfaccia costituita da un modulo contenente una casella di testo per ogni informazione modificabile di un regolamento, ed un bottone per la modifica.
- **GUIPubblicaRegolamento:** è l'interfaccia costituita da un modulo contenente un bottone per permettere all'utente di pubblicare un regolamento precedentemente scritto.

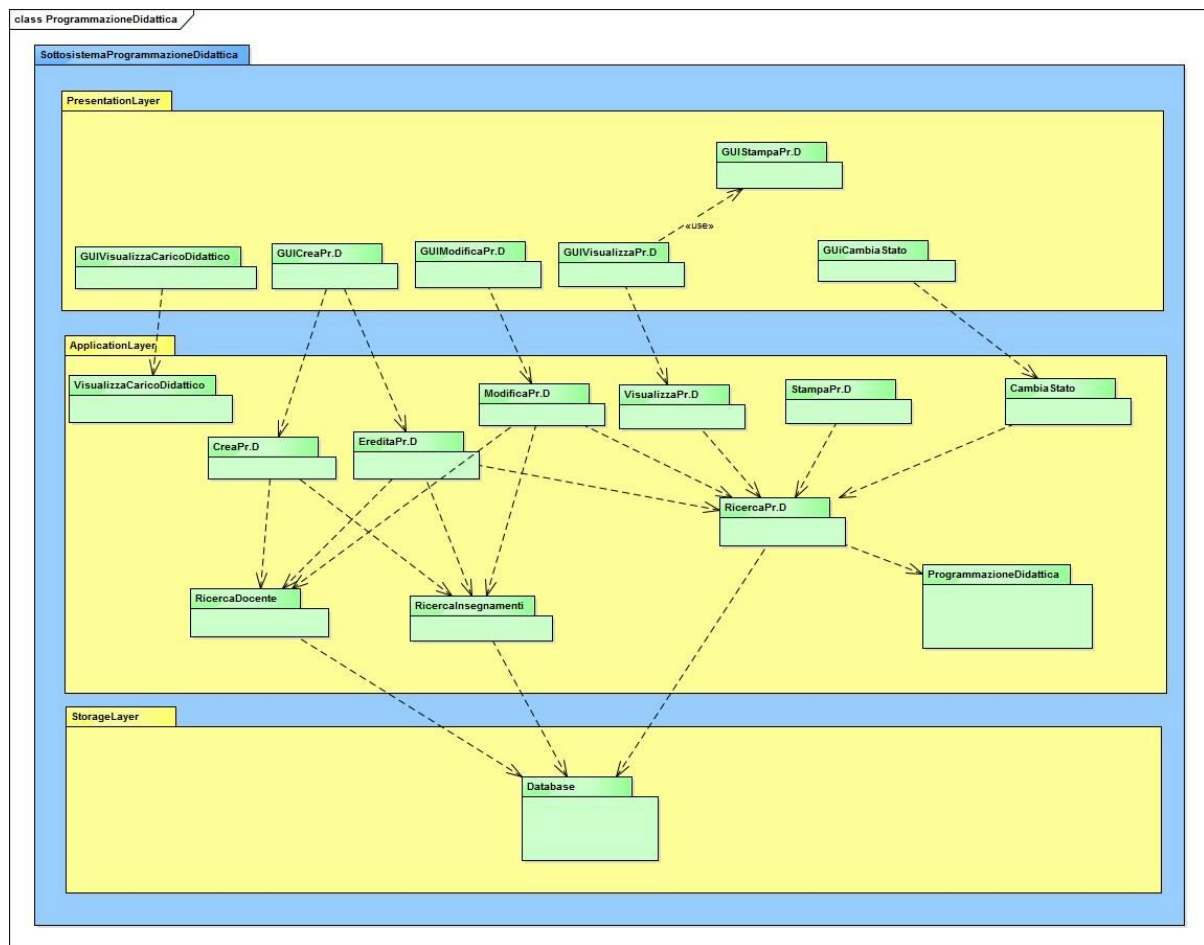
ApplicationLayer:

- **AggiungiRegolamento:** Si occupa di effettuare l'operazione di inserimento di un regolamento, interagendo con il database per verificare la correttezza dei dati.
- **VisualizzaRegolamento:** Si occupa mostrare all'utente il regolamento da visualizzare, cercandolo nel database secondo i dati inseriti dall'utente.
- **ModificaRegolamento:** Si occupa di effettuare il cambiamento dei dati di un dato regolamento e di comunicarli al database.
- **EreditaRegolamento:** Si occupa di ereditare un regolamento, già esistente nel database, e di reinserirlo con le relative modifiche.
- **PubblicaRegolamento:** Si occupa di pubblicare un regolamento precedentemente creato dall'utente.
- **RicercaRegolamento:** Si occupa di ricercare un regolamento presente all'interno del database.
- **Regolamento:** si occupa della gestione di tutte le operazioni riguardanti i regolamenti.

StorageLayer:

- **Database:** è il database contenente tutti i regolamenti presenti.

3.2.7 SubD_5 per Sottosistema gestione programmazione didattica



SottosistemaGestioneProgrammazioneDidattica, include tutte le componenti che permettono al Presidente di eseguire le operazioni di creazione, modifica e visualizzazione, mentre l'unica operazione consentita all'utente è quella di visualizzazione.

Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIVisualizzaCaricoDidattico**: è un'interfaccia che permette a un docente di visualizzare il proprio carico didattico e al Presidente di visualizzare il carico didattico degli altri docenti.
- **GUICreaPr.D.**: è un'interfaccia che consente di creare una nuova programmazione didattica e di aggiungerla al sistema.
- **GUIModificaPr.D.**: è un'interfaccia che consente di modificare una data programmazione didattica.
- **GUIVisualizzaPr.D.**: è un'interfaccia che consente di visualizzare una data programmazione didattica.

- **GUIStampaPr.D.:** è un'interfaccia che consente di stampare in formato PDF una data programmazione didattica.
- **GUICambiaStato:** è un'interfaccia che permette di cambiare lo stato di una data programmazione didattica.

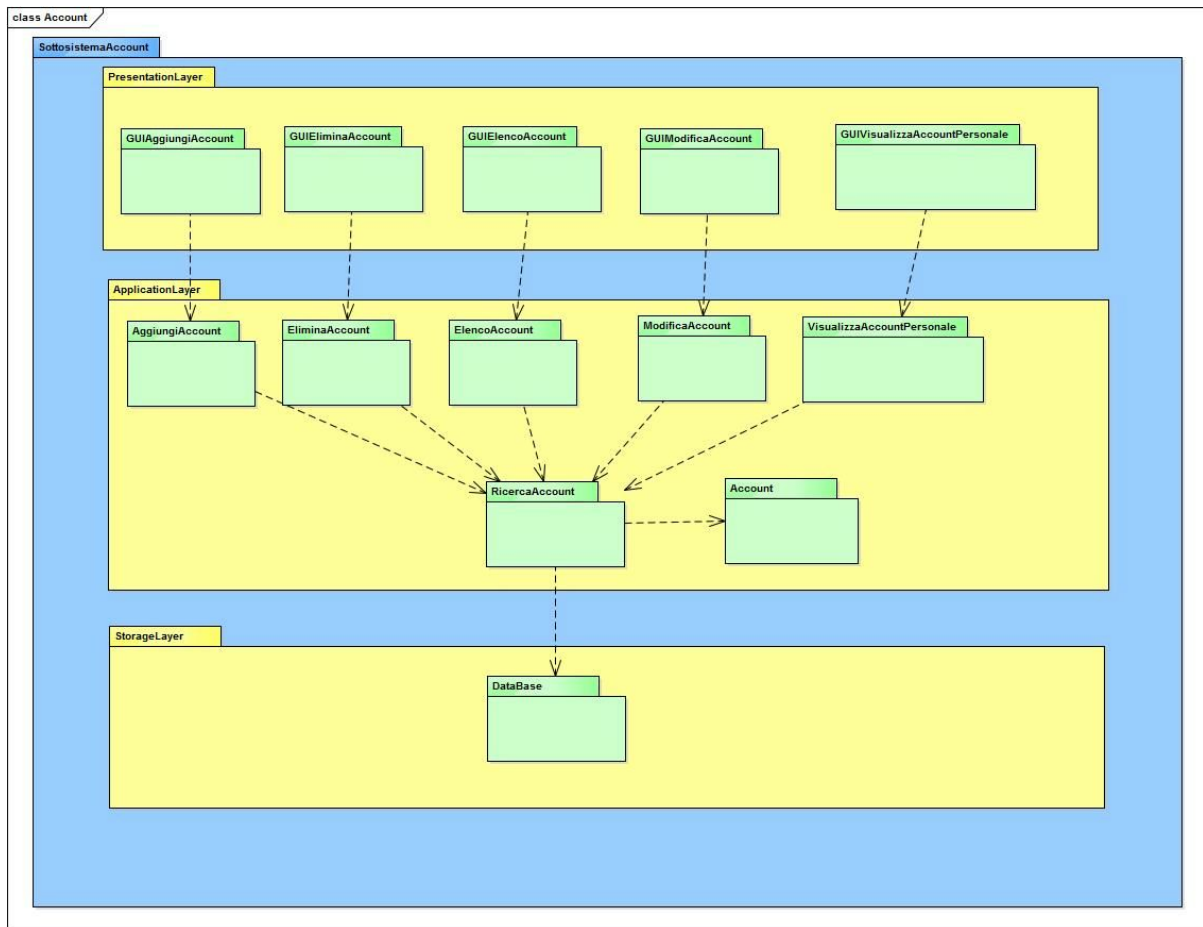
ApplicationLayer:

- **RicercaDocente:** permette di eseguire la ricerca di un docente.
- **RicercaPr.D:** permette di eseguire la ricerca di una programmazione didattica.
- **VisualizzaCaricoDidattico:** permette a un docente presente nel database e in possesso di un account di visualizzare il proprio carico didattico.
- **CreaPr.D:** permette di creare una nuova programmazione didattica.
- **EreditaPr.D:** permette di creare una nuova programmazione didattica ereditandola.
- **ModificaPr.D:** permette di modificare una data programmazione didattica.
- **VisualizzaPr.D:** permette di visualizzare una data programmazione didattica.
- **StampaPr.D:** permette di stampare in formato PDF una programmazione didattica presente nel database.
- **VisualizzaMonteOre:** permette di visualizzare il carico didattico del docente per cui è stata eseguita la ricerca.
- **CambiaStato:** permette al Presidente di cambiare lo stato di una programmazione didattica presente nel database e non ancora archiviata.
- **ProgrammazioneDidattica:** si occupa della gestione di tutte le operazioni riguardanti le programmazioni didattiche.

StorageLayer:

- **Database:** è il database in cui vengono memorizzate tutte le informazioni relative alla Programmazione Didattica.

3.2.8 SubD_6 per Sottosistema gestione account



SottosistemaGestioneAccount, include tutte le componenti che permettono al Presidente di eseguire le operazioni di aggiunta, modifica, visualizzazione ed eliminazione di un account mentre le operazioni consentite al docente sono quelle di visualizzazione e modifica. Di seguito è riportata una breve descrizione di ognuna delle componenti.

PresentationLayer:

- **GUIAggiungiAccount:** è un'interfaccia che permette al Presidente di aggiungere un nuovo account a un docente.
- **GUIEliminaAccount:** è un'interfaccia che permette al Presidente di eliminare un account appartenente a un docente.
- **GUIElencoAccount:** è un'interfaccia che permette al Presidente di visualizzare l'elenco degli account di tutti i docenti.
- **GUIModificaAccount:** è un'interfaccia che permette al Presidente e a un docente di modificare un account appartenente a un docente.
- **GUIVisualizzaAccountPersonale:** è un'interfaccia che permette al Presidente e a un docente di visualizzare il proprio account personale.

ApplicationLayer:

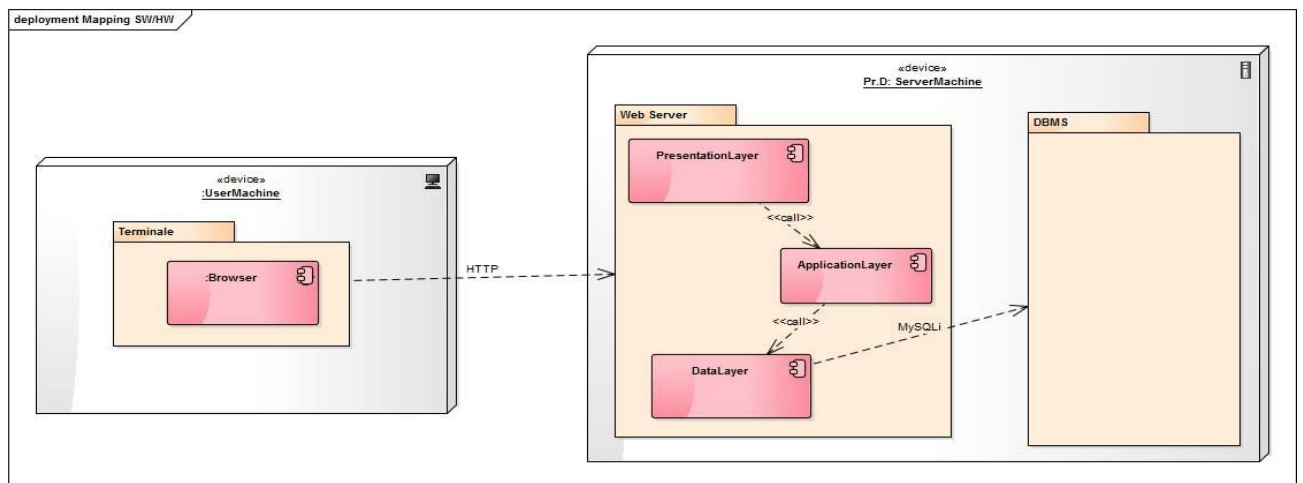
- **AggiungiAccount:** permette di aggiungere al database un nuovo account.
- **EliminaAccount:** permette di eliminare dal database un account esistente
- **ElencoAccount:** permette di visualizzare l'elenco di tutti gli account presenti nel database.
- **ModificaAccount:** permette di modificare un account presente nel database appartenente a un docente o al Presidente stesso.
- **VisualizzaAccountPersonale:** permette di visualizzare il proprio account personale.
- **RicercaAccount:** permette la ricerca di un account presente nel database.
- **Account:** si occupa della gestione di tutte le operazioni riguardanti gli account.

StorageLayer:

- **Database:** è il database in cui vengono memorizzate tutte le informazioni relative agli Account.

3.3 Hardware/software Mapping

Il sistema Pr.D è ideato per un tipo di architettura Client/Server. L'utente può accedere al nostro sistema tramite l'utilizzo di un browser mentre il sistema vero e proprio risiede all'interno di una macchina server. I terminali sono punti di appoggio per i vari utenti per poter accedere al sistema in modo da richiedere vari servizi al server, in base alle politiche di accesso. L'interfaccia utente cambia in base al tipo di accesso: Il presidente ha più funzionalità rispetto al Docente, e questo a sua volta ha più funzionalità di un utente. Con questo stile di mapping hardware e software il sistema ha una disponibilità illimitata in quanto i dati sul server sono disponibili 24 ore su 24.



Web Server	Il Web Server utilizzato è Apache Tomcat installato su una macchina server con sistema operativo Windows 10
PresentationLayer	Lo strato di “PresentationLayer” è lo strato di interfacciamento con l’utente, il quale verrà implementato nel Web Server e risiederà all’interno della ServerMachine. I contenuti realizzati con PHP, HTML, JS, AJAX, CSS verranno visualizzati mediante la componente “Browser” locata nella UserMachine
ApplicationLayer	Questo strato si occupa della logica di controllo del sistema. Lo scopo principale di questo strato è quello di fare da intermediario tra l’utente che interagisce solo con l’interfaccia del sistema e il livello che si occupa della conservazione dei dati.
DataLayer	Il DataLayer è rappresentato da un insieme di funzioni create in base alle esigenze date dai moduli specificati nello strato superiore del sistema. Questa scelta vuole generare una completa coesione fra il database fisico e le funzionalità che necessitano dell’accesso ai dati persistenti. La tecnologia che deve essere utilizzata per interfacciarsi con il sistema è l’estensione di PHP chiamata MySQLi.
DBMS	Dato il gran numero di accessi che è possibile avere all’interno del sistema in questione si è deciso di utilizzare un database relazionale realizzato con tecnologia DBMS MySQL, per memorizzare le informazioni utili al sistema. Il database è memorizzato all’interno della ServerMachine che ospita l’intero sistema.

3.4 Gestione dati persistenti

Fare riferimento al file [Pr.D_DBD_2.0](#)

3.5 Controllo degli accessi e sicurezza

Gli utenti che operano nel sistema non possono avere libero accesso ad ogni funzionalità presente. Da questo si deduce che l'accesso a ogni elemento del sistema è regolato e prestabilito per evitare modifiche indesiderate, malfunzionamenti e utilizzo di parti del sistema riservati ad altri utenti. L'unico utente che ha il completo accesso al sistema e ha la possibilità di modificare informazioni nei database tramite login, è il Presidente, mentre i docenti hanno accesso a ridotte funzionalità rispetto ad esso.

Di seguito sono riportate le tabelle degli accessi del sistema Pr.D.

Oggetto Attori	Autenticazione	Docenti	Insegnamenti	Ordinamento	Regolamento	Programmazione Didattica	Account
Presidente	<<Login>> <<Logout>>	<<Aggiungi>> <<Visualizza>> <<Modifica>> <<Stato>>	<<Aggiungi>> <<Visualizza>> <<Modifica>>	<<Visualizza>>	<<Crea>> <<Modifica>> <<Pubblica>> <<Visualizza>>	<<Crea>> <<Visualizza>> <<Visualizza Monte ore>> <<Stato>> <<Stampa>>	<<Aggiungi>> <<Elimina>> <<Visualizza>> <<Elenco>>
Docente	<<Login>> <<Logout>>	<<Visualizza>>	<<Visualizza>>	<<Visualizza>>	<<Visualizza>>	<<Visualizza>> <<Visualizza Carico didattico>> <<Stampa>>	<<Visualizza>>
Utente		<<Visualizza>>	<<Visualizza>>	<<Visualizza>>	<<Visualizza>>	<<Visualizza>> <<Stampa >>	

<<Login>>	Permette l'accesso al sistema
<<Logout>>	Permette l'uscita dal sistema.
<<Aggiungi>>	Crea una nuova istanza dell'oggetto
<<Elimina>>	Elimina l'oggetto selezionato
<<Visualizza>>	Visualizza l'oggetto attraverso una ricerca, ma non permette di modificarlo
<<Modifica>>	Modifica un oggetto esistente
<<Pubblica>>	Pubblica l'oggetto definitivamente rendendo visibile a tutti gli utenti
<<Stato>>	Cambia lo stato dell'oggetto
<<Stampa>>	Stampa l'oggetto selezionato
<<Elenco>>	Visualizza un elenco di oggetti

3.6 Gestione del controllo globale

Il sistema Pr. D. utilizzerà un controllo del flusso event-driven dato che prevede che il flusso del programma è largamente determinato dal verificarsi di eventi esterni.

Pr.D. è un sistema costituito in modo che ogni funzionalità viene avviata solo dopo un comando dell'utente.

Il sistema rimane quindi in attesa di una richiesta da parte dell'utente, il quale, tramite la pressione di un bottone o tramite la scelta di un'opzione dal menu scatena un evento, che sarà gestito correttamente dal giusto handler, che indirizzerà il controllo del flusso del sistema alla sezione corretta del sottosistema che si occupa della logica di controllo.

3.7 Condizioni Boundary

All'avvio agli utenti si presenta un'interfaccia intuitiva dalla quale è possibile accedere solo a un numero limitato di operazioni. Queste operazioni sono quelle riservate all'utente generico che accede al sistema senza effettuare l'accesso. Il Presidente e i Docenti sono gli unici utenti ad avere la possibilità di effettuare il login. Ad autenticazione avvenuta il Presidente avrà pieno accesso a tutte le funzionalità disponibili, mentre i Docenti avranno accesso ad un numero di funzionalità ridotte.

Nel caso in cui si verifichi un crash del sistema o una mancanza improvvisa di connessione, i dati vengono comunque salvati nel Database. L'utilizzo di un DBMS e della tecnologia MySQL elimina il rischio di perdita di dati. Solo nel caso in cui si verificano dei guasti nei dati, i dati potrebbero andare persi.

In caso di crash dovuto a un bug nel codice, si distinguono 2 casi:

- Il crash si verifica solamente nel client. Il sistema può continuare a operare regolarmente
- Il crash si verifica nel Server. Il sistema risulta inutilizzabile in quanto non è possibile stabilire la connessione con il server.

3.7.1 Scenario

Questo scenario rappresenta la situazione in cui il server non è ancora stato avviato, quindi, si prevede che l'Amministratore esterno possa avviare il servizio utilizzando l'interfaccia grafica messa a disposizione.

3.7.1_1 Startup del Server

SC_BC_1	Startup del Server
Partecipanti	L'Amministratore esterno
Flusso di Eventi	<ul style="list-style-type: none">● L'Amministratore esterno , dopo aver effettuato l'accesso al sistema, richiede l'avvio del server tramite l'opportuno comando per mettere a disposizione remota i servizi offerti.● Il sistema mette a disposizione i servizi in remoto e si mette in attesa di richieste.● Il sistema restituisce conferma dell'operazione.

Nel caso successivo invece l'Amministratore Esterno vuole arrestare il server. Anche in questa situazione il Presidente può utilizzare l'interfaccia grafica offerta dal sistema.

3.7.1_2 Shoutdown del Server

SC_BC_2	Shoutdown del Server
Partecipanti	L'Amministratore esterno
Flusso di Eventi	<ul style="list-style-type: none">● L'Amministratore esterno dopo aver effettuato l'accesso al sistema, richiede l'arresto del server tramite l'opportuno comando.● Il sistema porta a termine le operazioni in esecuzione.● Il sistema rimuove i servizi offerti in remoto e ne dà conferma.

3.7.2 Casi d'uso

3.7.2_1 Startup del Server

UC_BC_3	Startup del Server
Partecipanti	Amministratore Esterno
Flusso di Eventi	<ul style="list-style-type: none">● Amministratore Esterno : Richiede il lancio del server.● Sistema: Effettua l'avvio del server, mettendo disponibili in remoto le varie funzionalità offerte.
Condizioni d'ingresso	<ul style="list-style-type: none">● Il Amministratore Esterno ha effettuato l'accesso al sistema.● Il Amministratore Esterno deve avviare i servizi del server.
Condizioni di uscita	<ul style="list-style-type: none">● I servizi sono resi disponibili in remoto ai client ed il sistema conferma l'esito all'Amministratore Esterno .
Eccezioni	<ul style="list-style-type: none">● Il Amministratore Esterno annulla l'operazione e torna indietro.
Requisiti di qualità	<ul style="list-style-type: none">● L'avvio del server avviene in tempi brevi.

3.7.2_2 Shutdown del Server

UC_BC_4	Shutdown del Server
Partecipanti	Amministratore Esterno
Flusso di Eventi	<ul style="list-style-type: none">● Amministratore Esterno : Richiede l'arresto del server.● Sistema: Effettua l'arresto del server, rimuovendo i servizi resi disponibili in remoto.
Condizioni d'ingresso	<ul style="list-style-type: none">● Il Amministratore Esterno ha effettuato l'accesso al sistema.● Il Amministratore Esterno deve avviare i servizi del server.
Condizioni di uscita	<ul style="list-style-type: none">● I servizi remoti vengono rimossi ed il server viene arrestato; il sistema conferma l'esito al Amministratore Esterno .
Eccezioni	<ul style="list-style-type: none">● Il Amministratore Esterno annulla l'operazione e torna indietro.
Requisiti di qualità	<ul style="list-style-type: none">● L'arresto del server avviene in tempi brevi.

4 Servizi dei sottosistemi

4.1 SS_0 Sottosistema per RF_0 Gestione Autenticazione

Application Layer	Presentation Layer	Storage Layer
Login	GUIPresidente -> GUILogin	//
Logout	GUIPresidente -> GUILogout	//
RecuperaPassword	GUIPresidente -> GUIRecuperaPassword	//
Autenticazione	//	Database

4.2 SS_1 Sottosistema per RF_1 Gestione Docenti

Application Layer	Presentation Layer	Storage Layer
AggiungiDocenti	GUIPresidente -> GUIAggiungiDocenti	//
VisualizzaElencoDocenti	GUIPresidente/Utente -> GUIVisualizzaElencoDocenti	//
VisualizzaDettagliDocente	GUIUtente -> GUIVisualizzaDettagliDocenti	//
RicercaDocenti	//	//
ModificaInformazioni	GUIPresidente/Docente -> GUIModificaInformazioni	//
ModificaStato	GUIPresidente/Docente -> GUIModificaStato	//
Docente	//	Database

4.3 SS_2 Sottosistema per RF_2 Gestione Insegnamenti

Application Layer	Presentation Layer	Storage Layer
AggiungiInsegnamento	GUIPresidente -> GUIAggiungiInsegnamenti	//
VisualizzaInsegnamenti	GUIUtente -> GUIVisualizzaInsegnamenti	//
RicercaInsegnamenti	//	//
ModificaInformazioniInsegnamento	GUIPresidente -> GUIModificaInformazioniInsegnamento	//
Insegnamento	//	Database

4.4 SS_3 Sottosistema per RF_3 Gestione Ordinamento

Application Layer	Presentation Layer	Storage Layer
VisualizzaOrdinamento	GUIUtente -> GUIVisualizzaOrdinamento	Database

4.5 SS_4 Sottosistema per RF_4 Gestione Regolamento

Application Layer	Presentation Layer	Storage Layer
VisualizzaRegolamento	GUIUtente -> GUIVisualizzaRegolamento	//
AggiungiRegolamento	GUIPresidente -> GUIAggiungiRegolamento	//
EreditaRegolamento	//	//
ModificaRegolamento	GUIPresidente -> GUIModificaRegolamento	//
RicercaRegolamento	//	//
PubblicaRegolamento	GUIPresidente -> GUIPubblicaRegolamento	//
Regolamento	//	Database

4.6 SS_5 Sottosistema per RF_5 Gestione Programmazione Didattica

Application Layer	Presentation Layer	Storage Layer
VisualizzaCaricoDidattico	GUIDocente -> GUIVisualizzaCaricoDidattico	//
CreaProgrammazioneDidattica	GUIPresidente -> GUICreaProgrammazioneDidattica	//
EreditaProgrammazioneDidattica	//	//
ModificaProgrammazioneDidattica	GUIPresidente -> GUIModificaProgrammazioneDidattica	//
RicercaDocente	//	//
RicercaProgrammazioneDidattica	//	//
VisualizzaProgrammazioneDidattica	GUIUtente -> GUIVisualizzaProgrammazioneDidattica	//
VisualizzaMonteOreDocente	GUIPresidente -> GUIVisualizzaMonteOreDocente	//
CambiaStato	GUIPresidente -> GUICambiaStato	//
StampaPdf	GUIUtente -> GUIStampaPdf	//
ProgrammazioneDidattica	//	Database

4.7 SS_6 Sottosistema per RF_6 Gestione Account

Application Layer	Presentation Layer	Storage Layer
AggiungiAccount	GUIPresidente -> GUIAggiungiAccount	//
EliminaAccount	GUIPresidente -> GUIEliminaAccount	//
VisualizzaElencoAccount	GUIPresidente -> GUIVisualizzaElencoAccount	//
ModificaAccount	GUIUtenteRegistrato -> GUIModificaAccount	//
VisualizzaAccountPersonale	GUIUtenteRegistrato -> GUIVisualizzaAccountPersonal e	//
RicercaAccount	//	//
Account	//	Database

5 Glossario

Per una perfetta comprensione dell'SDD da parte di qualsiasi operatore poco esperto, vengono di seguito riportati acronimi e abbreviazioni utilizzate nel documento.

1. **RAD**: Requirements Analysis Document

2. **SDD** : System Design Document

3. **DBMS**: Database Management System

4. **DB**: DataBase

5. **Pr. D.**: Nome del Sistema Software

6. **HW**: Hardware

7. **SW**: Software

8. **SubD**: Subsystem design

9. **DG**: Design goal