UNIVERSITÀ DEGLI STUDI DI SALERNO

INGEGNERIA DEL SOFTWARE

Integration Test Plan (ITP)

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
02/12/2015	1.0	Compilazione informazioni test d'integrazione	Gianmaria Trezza Stefano Cirillo
22/12/2015	1.1	Revisione ITP	Alberto Sergio
22/01/2015	1.2	Controllo ITP	Alessandro Barletta

Sommario

1 INTRODUZIONE

Il testing di integrazione rappresenta una delle fasi di testing più importanti, consiste nella verifica delle interazioni tra due o più componenti che hanno superato i System Test.
L'obiettivo del testing consiste nella **verifica della corretta interazione tra le componenti** e della congruenza dei contenuti, secondo quanto stabilito nelle Specifiche di Integrazione.

Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema **Pr.D.**

2 RELAZIONE CON GLI ALTRI DOCUMENTI

Per verificare la corretta integrazione dei sottosistemi del sistema **Pr.D.** sono stati predisposti dei test case basati sulla divisione in sottosistemi in fase di System Design.

Il documento di riferimento è: PrD_SDD_2.1-

3 DETTAGLI DEL LEVEL TESTING DESIGN

3.1 Approccio di Integration Testing

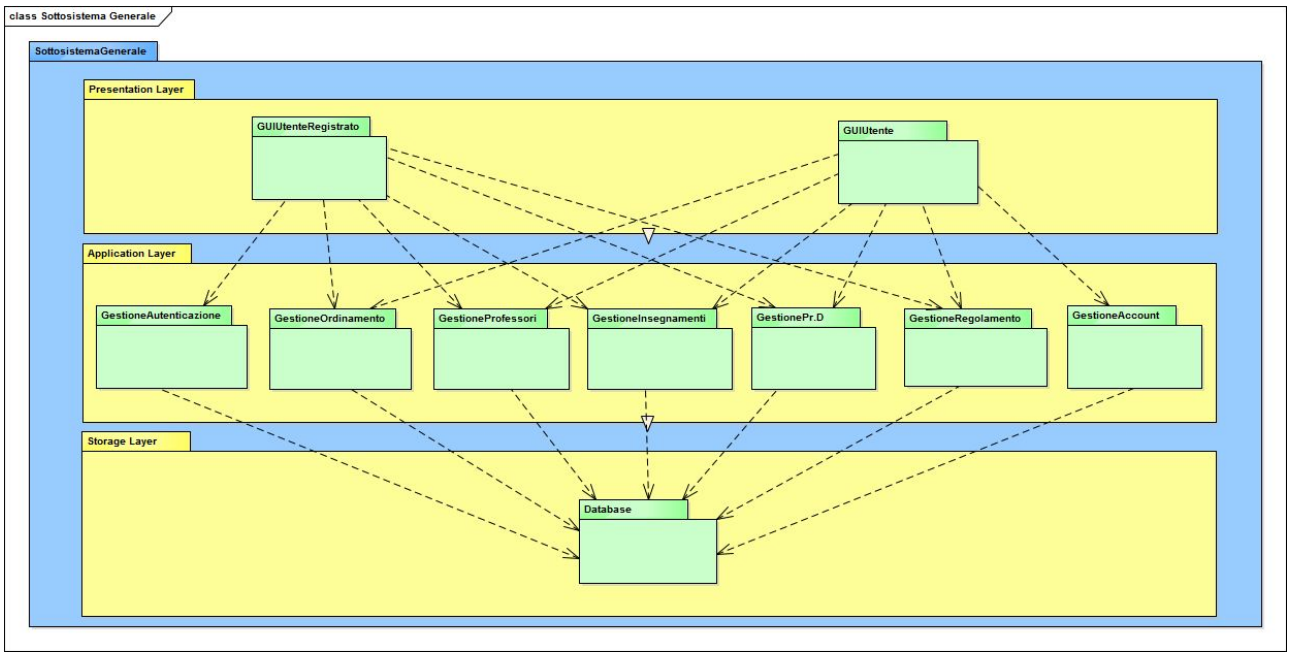
La strategia adottata per il testing di integrazione è quella di tipo “Bottom-up”.
La strategia adottata prevede che i sottosistemi dei layer di livello piu’ basso della gerarchia vengano testati individualmente; successivamente verranno testati i sottosistemi dei layer di livello superiore fino ad arrivare alle interfacce grafiche.
I **test driver** sono una implementazione parziale di una componente che dipende dalla componente testata e sono usati per simulare le componenti dei layer piu’ in alto che non sono stati ancora integrati.

Nonostante questa strategia abbia alcune limitazioni, risulta essere la piu’ semplice ed indicata con cui eseguire questo tipo di testing.

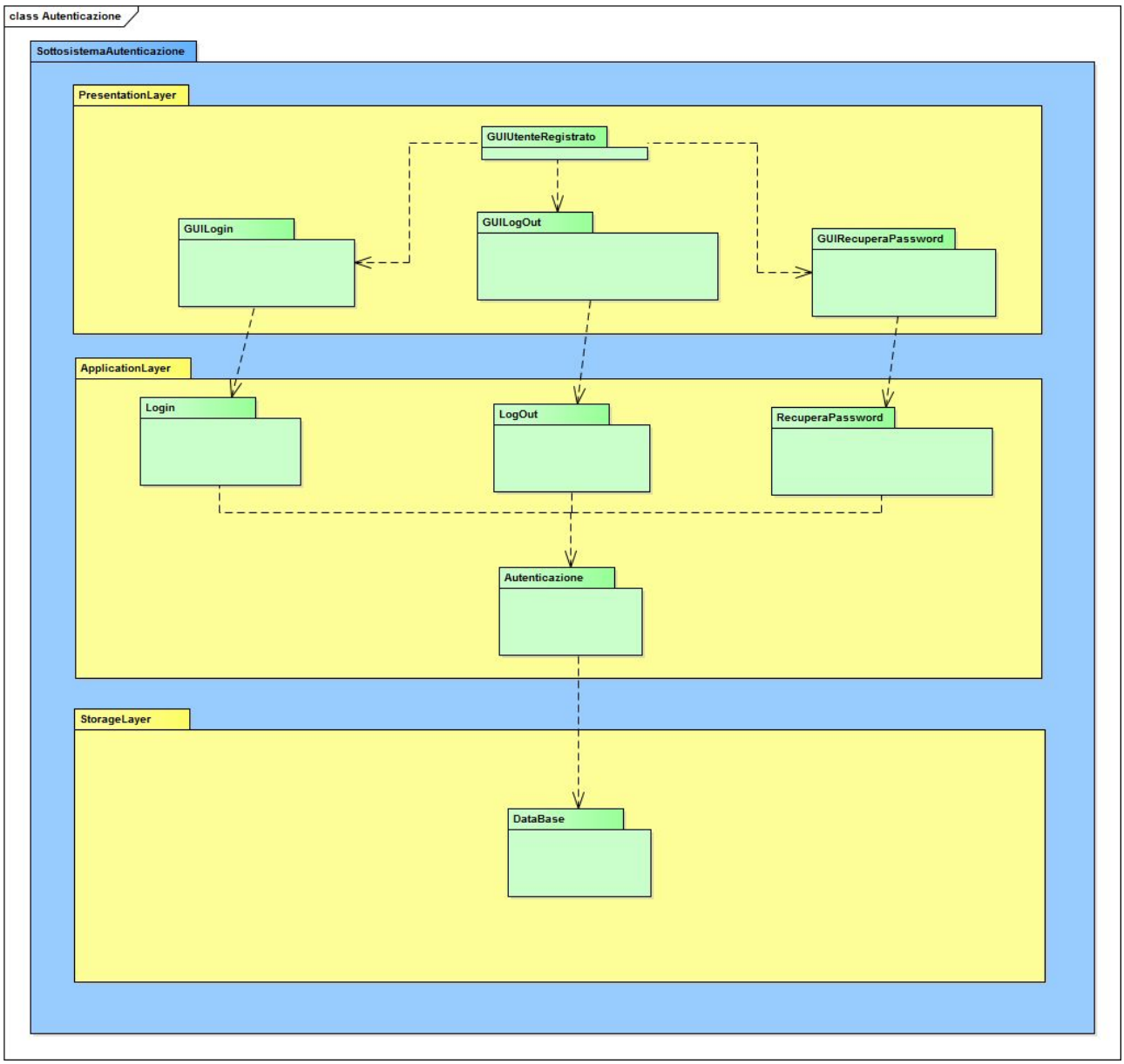
3.2 Componenti da testare

La scelta delle componenti da testare segue la decisione di eseguire la strategia di testing “Bottom-up”

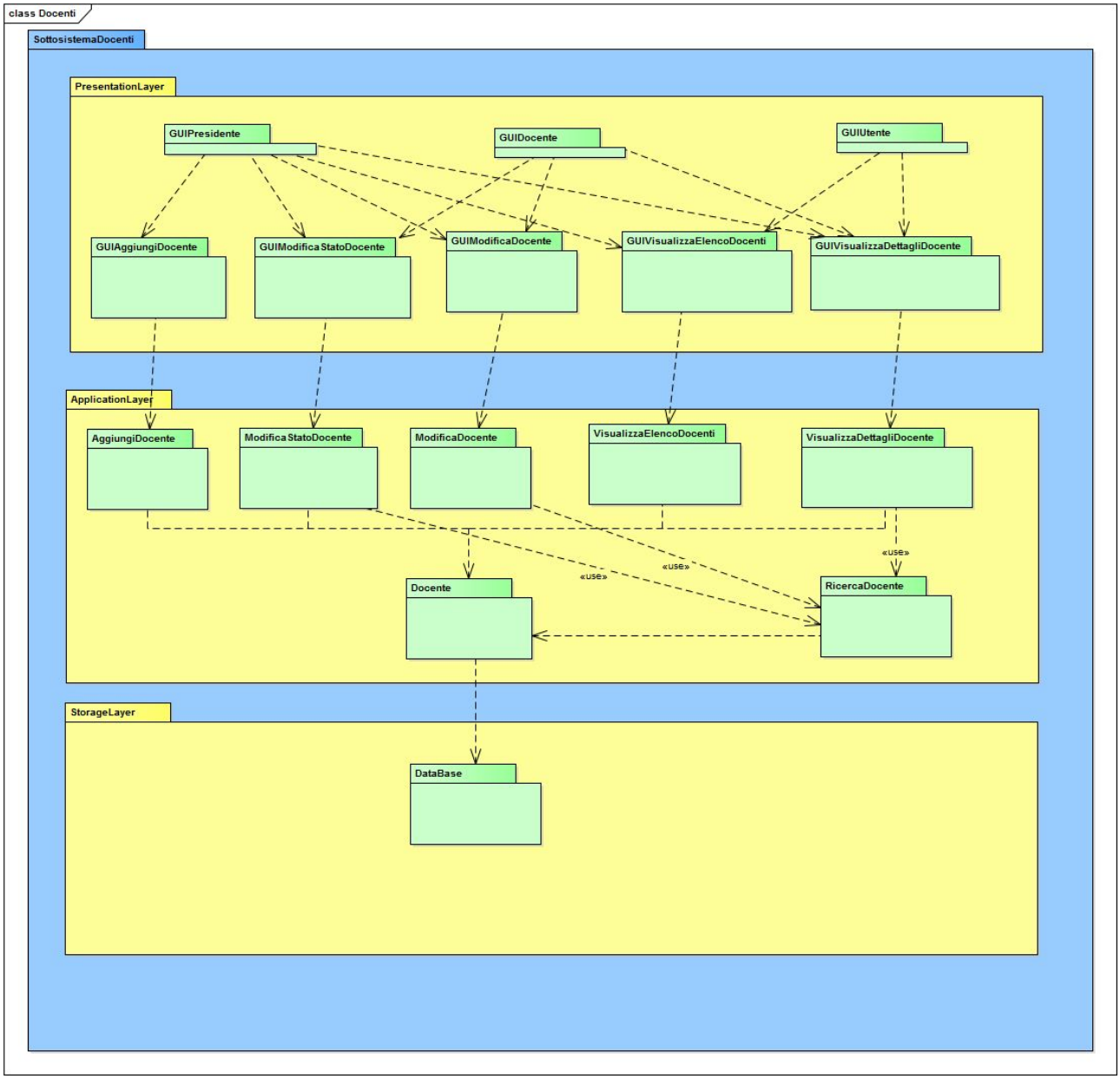
Sottosistema generale



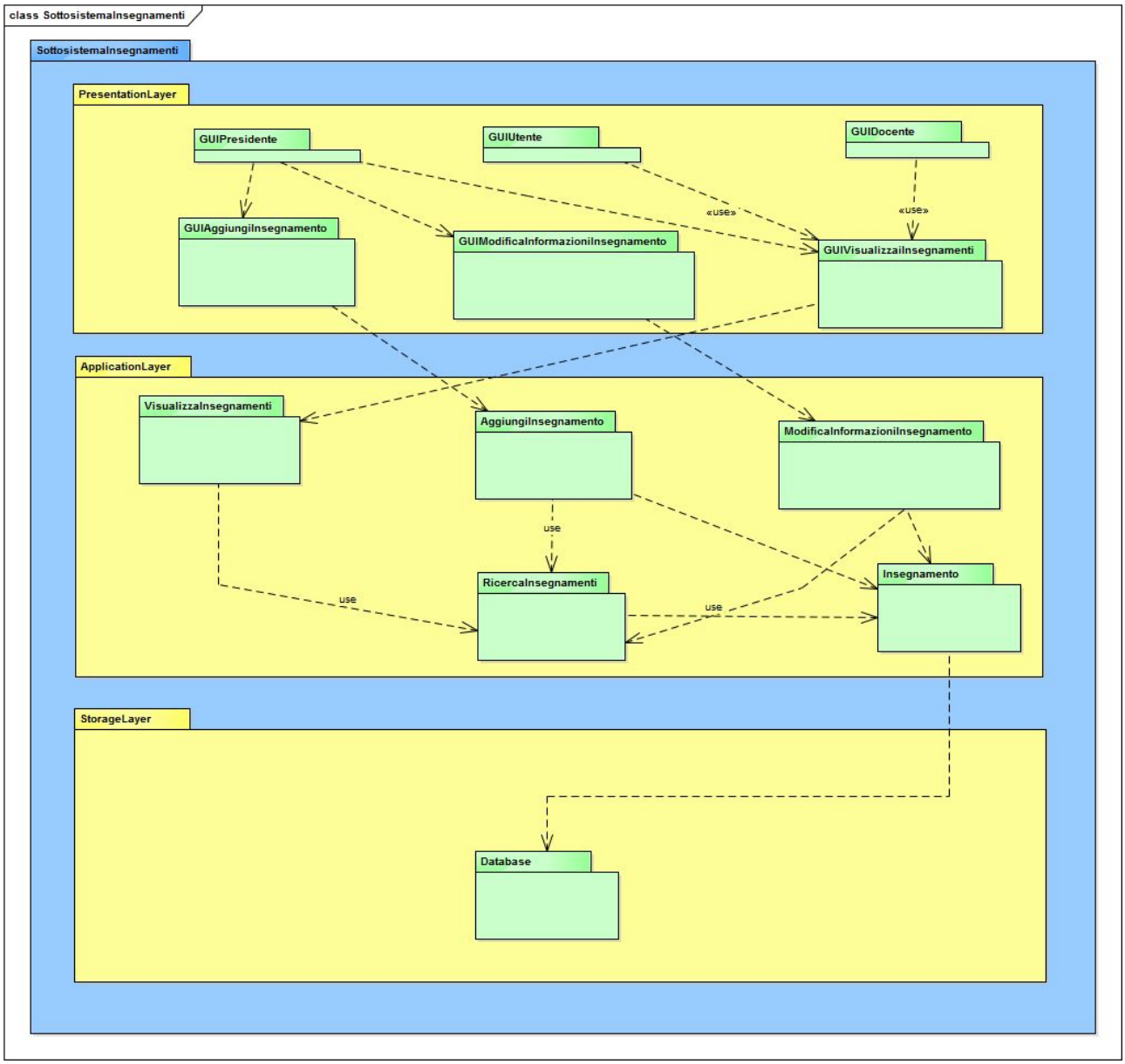
Sottosistema gestione Autenticazione



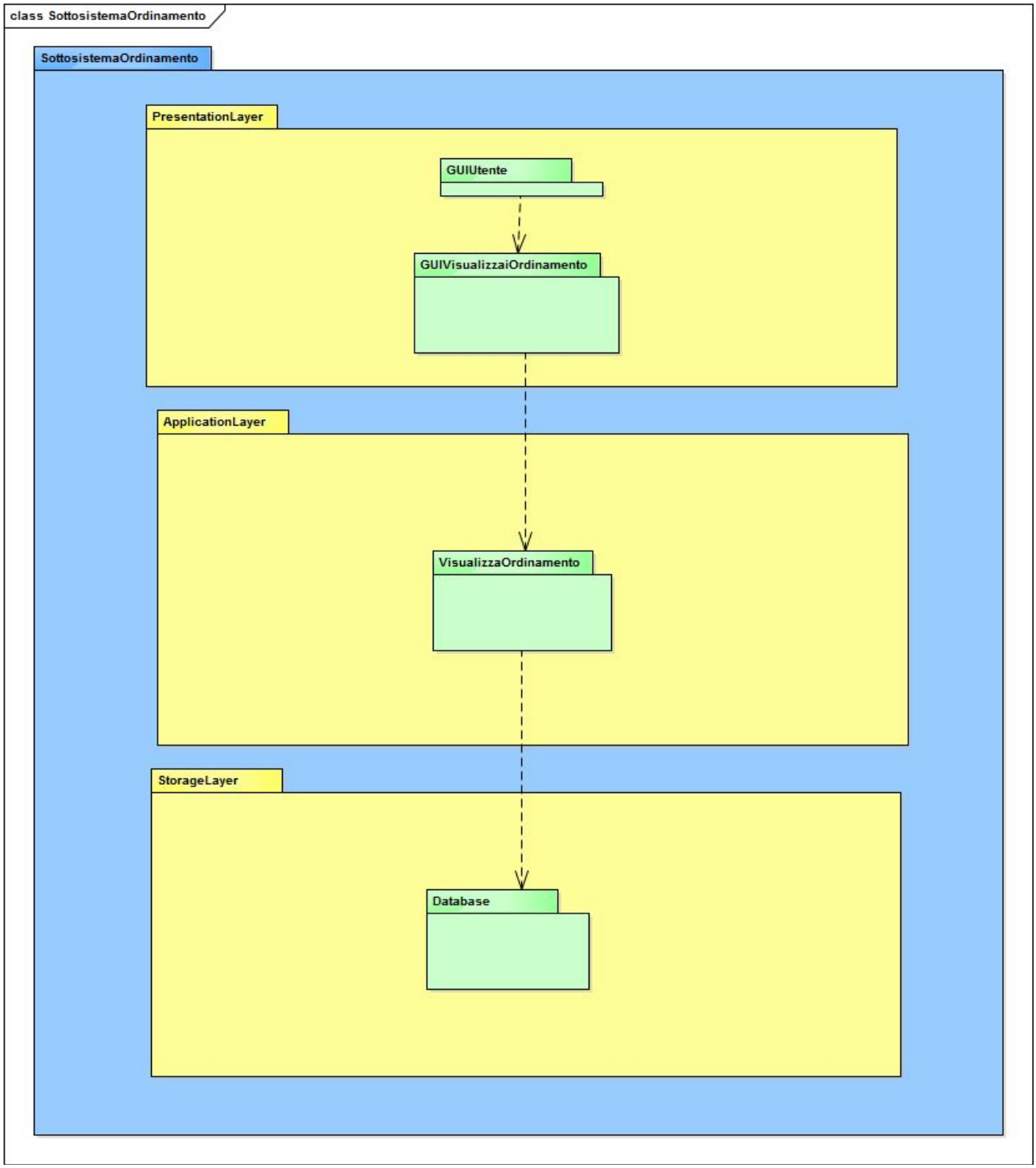
Sottosistema Docenti



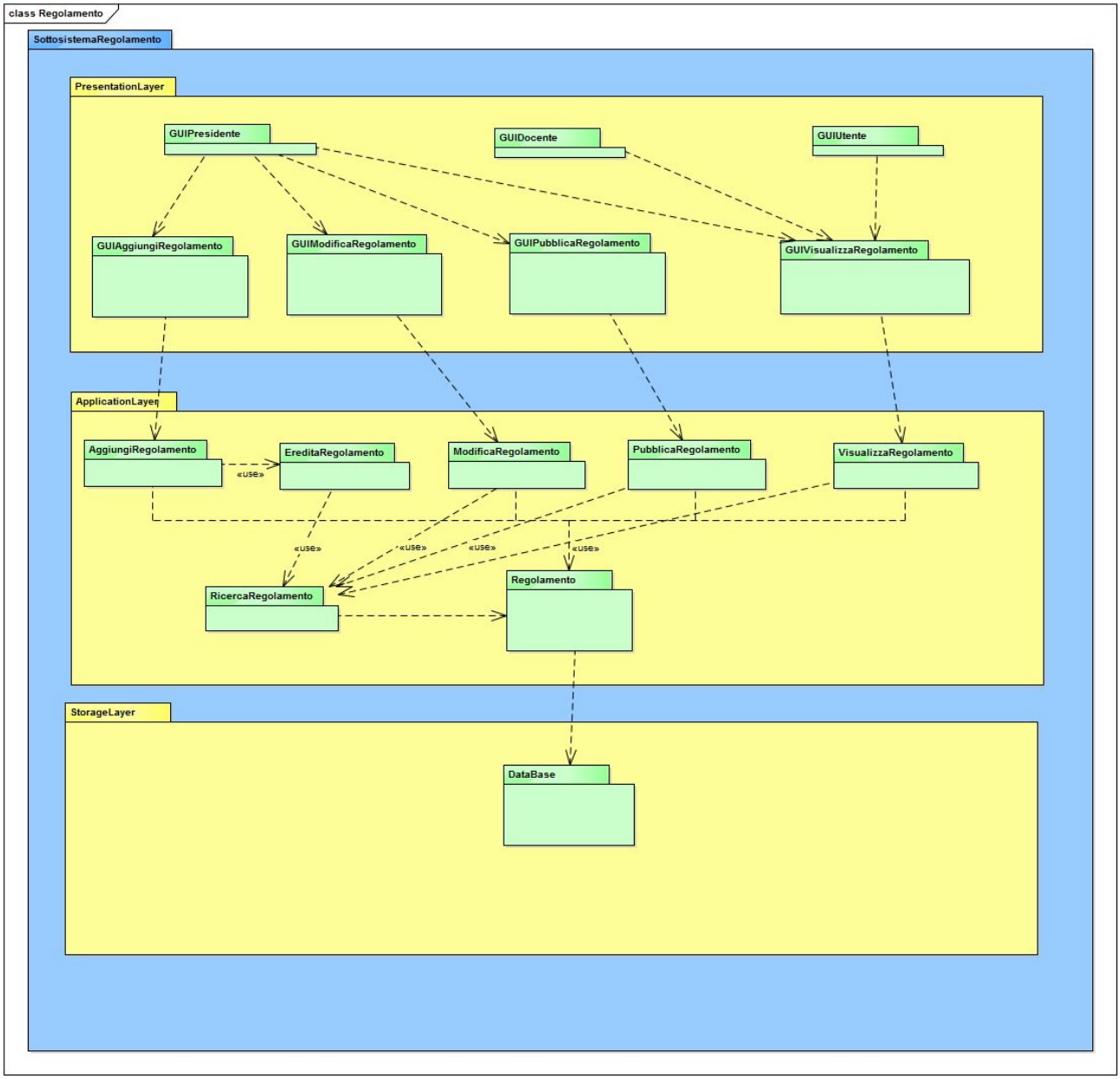
Sottosistema insegnamenti



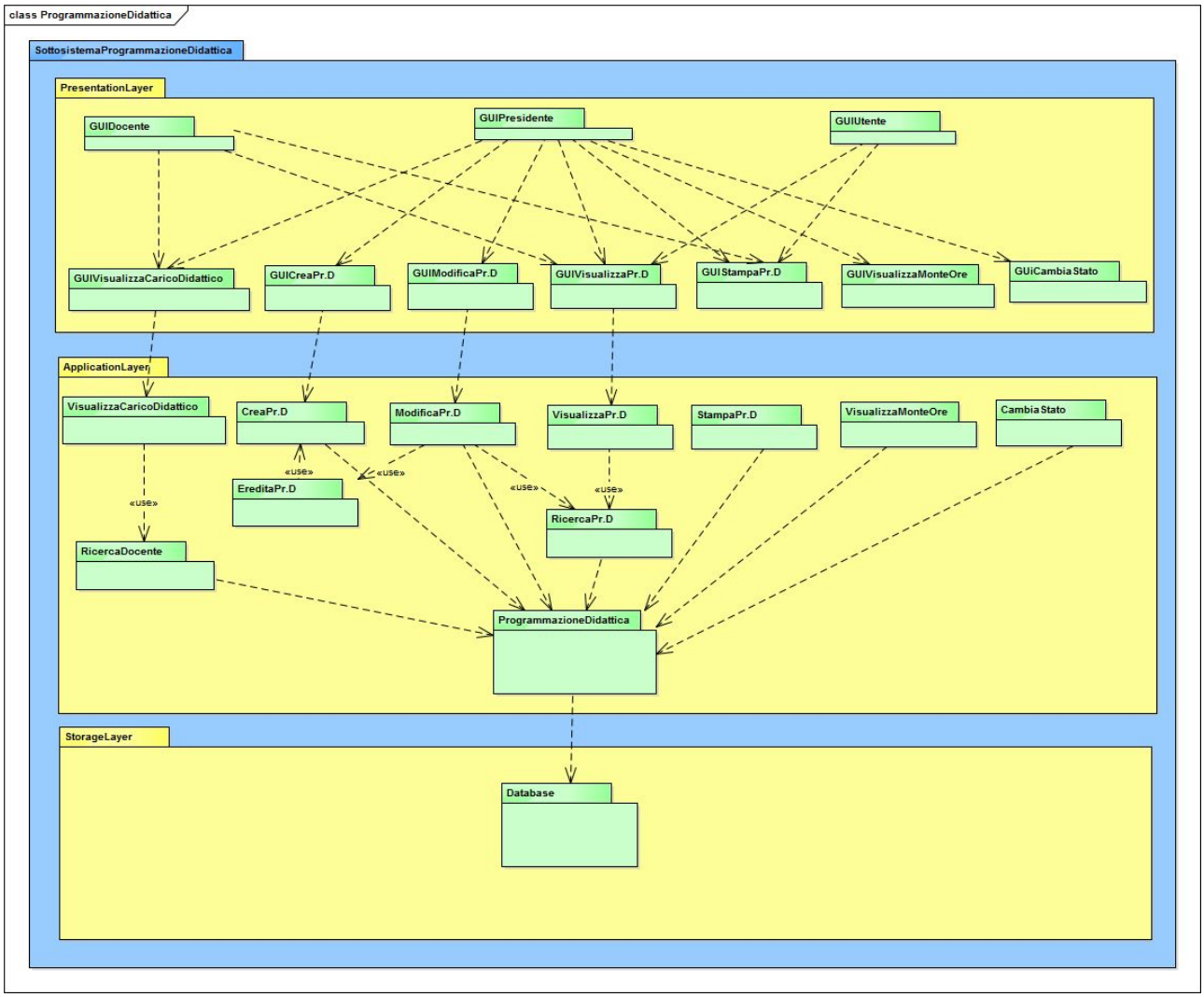
Sottosistema ordinamento



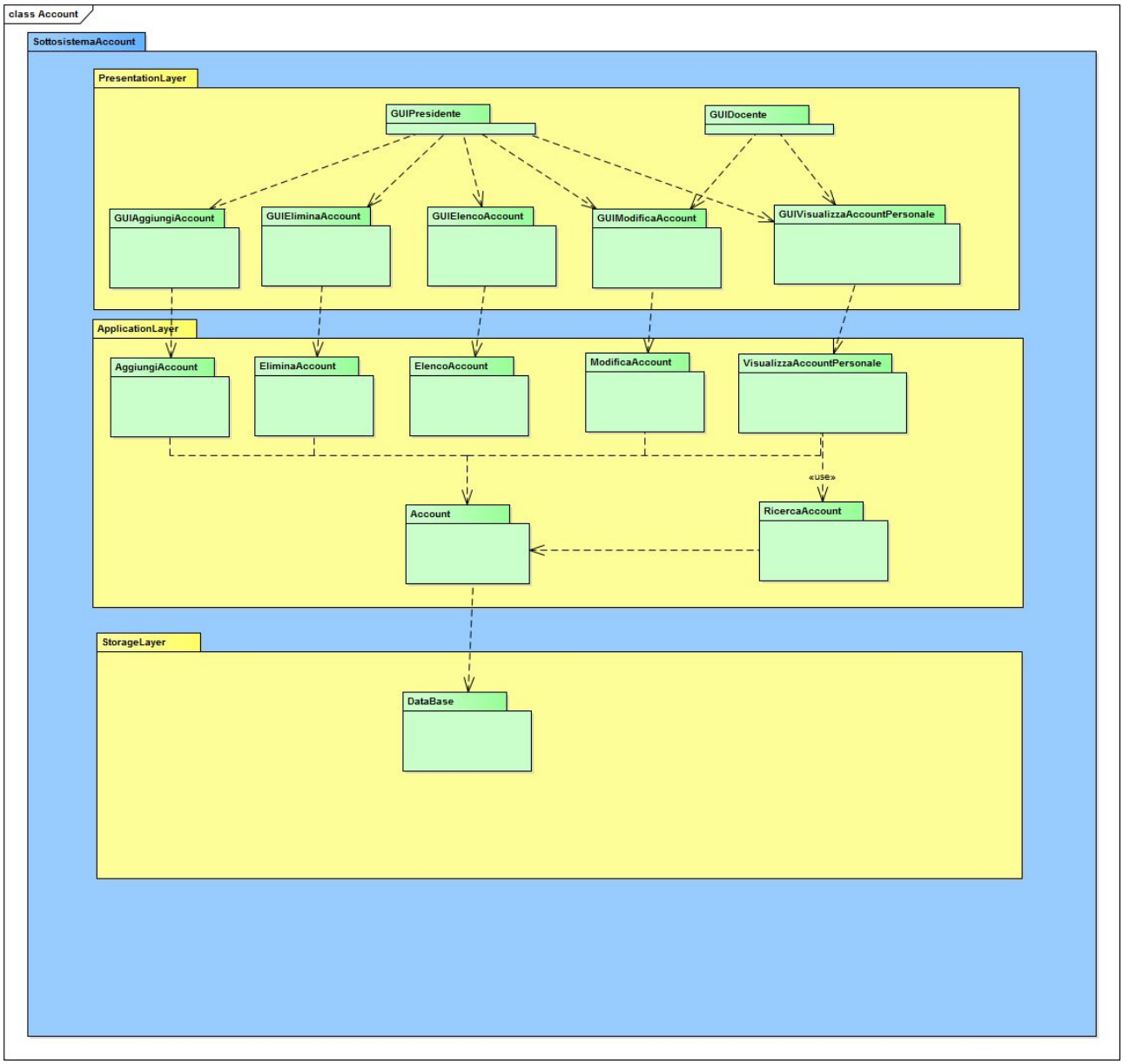
Sottosistema regolamento



Sottosistema programmazione didattica



Sottosistema Account



4 PASS/FAIL CRITERI

La verifica della corretta funzionalità di una componente si basa sugli input forniti al sistema e sugli output che questo restituisce. In questo modo si decreta se il test è stato superato o meno.

I dati inseriti in input sono suddivisi in classi di equivalenza. Una componente supera un test se e solo se l'output ottenuto in base ai dati in input è quello atteso.

Gli sviluppatori conoscono tutte le funzionalità che il sistema deve avere e il modo in cui questo si deve comportare. Sono state testate e verificate che tutte le funzionalità e i comportamenti del sistema rispondano ai requisiti richiesti.

Ogni funzionalità che è stata testata, è stata accuratamente analizzata e in base ai parametri di input, sono state specificate tutte le possibili combinazioni di utilizzo che potrebbero portare alla generazione di errori nel sistema.
Il Team al verificarsi di fallimenti nella fase di testing ha provveduto a prendere atto dell'evento ed a modificare opportunamente il codice.