

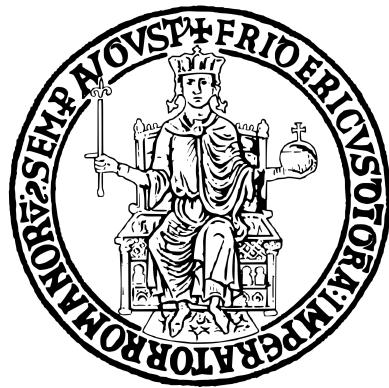
UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

OBJECT ORIENTATION

**Progettazione e sviluppo di una base
di dati relazionale per una
applicazione di e-learning**

Marzia PIROZZI N86003545
Noemi SPERA N86003717

Luglio 2022



Indice

1 DESCRIZIONE DEL PROGETTO	2
1.1 Identificazione delle classi e delle responsabilità	3
2 PROGETTAZIONE	5
2.1 Diagramma del dominio del problema	5
2.2 Diagramma del dominio della soluzione del problema	6
2.3 Sequence diagram	7
2.3.1 Svolgimento Test:	7
2.3.2 Creazione Test:	8
3 MANUALE D'USO	10

Capitolo 1

DESCRIZIONE DEL PROGETTO

Si sviluppi un applicativo Java dotato di GUI (Swing o JavaFX) per l'e-learning che consenta di gestire test basati su quiz. I test possono essere inseriti nel sistema da un insegnante, che si registrerà inserendo il proprio nome e cognome, oltre che una login e una password. Un test consiste di un insieme di quiz che deve essere fissato alla creazione del test. Ogni test è caratterizzato da un nome univoco che lo identifica. Esistono due tipi di quiz: quiz 'a risposta multipla' e quiz 'a risposta aperta'. Un quiz a risposta multipla è caratterizzato da una domanda (espressa tramite un breve testo), un elenco di possibili risposte (ognuna delle quali descritta da un breve testo ma una sola delle quali è da ritenersi corretta), il punteggio da assegnare in caso di risposta esatta e il punteggio (eventualmente anche negativo) da assegnare in caso di risposta errata. Un quiz a risposta aperta è caratterizzato, oltre che da un breve testo che descrive la domanda posta, dalla massima lunghezza prevista per il testo di risposta e dai punteggi minimo e massimo che l'insegnante potrà assegnare in base alla correttezza della risposta. Anche gli studenti si registreranno al sistema immettendo il proprio nome e cognome oltre che una login e password. Uno studente può sostenere un test scegliendolo dall'elenco dei test inseriti, fornendo una risposta tra quelle proposte per ogni quiz a risposta multipla e un testo per ogni quiz a risposta aperta. L'insegnante che ha creato il test ha il compito di valutare la correttezza di tutte le risposte ai quiz a risposta aperta degli studenti che hanno sostenuto il test assegnando un punteggio compreso tra il minimo e il massimo previsti per quel quiz, in modo che il sistema possa calcolare il numero esatto di risposte corrette fornite dallo studente per quel test.

1.1 Identificazione delle classi e delle responsabilità

Abbiamo identificato le seguenti classi:

Nome	Descrizione
Utente	Nome, Cognome, Login, Password dell'utente, cin responsabilità di effettuare il login e registrare un utente
Insegnante	Prende tutti gli attributi e le responsabilità di utente più altre sue responsabilità come la creazione di un test o la valutazione dei quiz a risposta aperta
Studente	Prende tutti gli attributi e le responsabilità di utente più altre sue responsabilità come lo svolgimento di un test o la visione dei test svolti corretti
Test	N_quiz Int: Numero di quiz che costituiscono il test Data Date: Data di svolgimento del test Orario_inizio Time: Orario di inizio per lo svolgimento del test Orario_fine Date: Orario di consegna del test
Quiz_Risp_ape	Domanda Varchar[100]: Testo della domanda di un quiz a risposta aperta Max_lunghezza Int: Lunghezza massima della risposta espressa in intero (esempio 100= risposta lunga 100 caratteri) Punt_max Real: Il punteggio che verrà assegnato se l'insegnante ritiene che la risposta sia completamente corretta Punt_min Real: Il punteggio minimo che può essere assegnato in caso di risposta errata o parzialmente corretta
Quiz_Risp_Mul	Domanda Varchar[100]: Testo della domanda di un quiz a risposta multipla Risposta A Varchar [20]: Testo della risposta A Risposta B Varchar [20]: Testo della risposta B Risposta C Varchar [20]: Testo della risposta C Risposta D Varchar [20]: Testo della risposta D Risposta_corretta Varchar[20]: Quale delle alternative è effettivamente la risposta corretta Punt_corretto Int: Il punteggio che verrà assegnato dal sistema se la risposta registrata dallo studente corrisponde a quella corretta Punt_errato Int: Il punteggio che verrà assegnato dal sistema se la risposta registrata dallo studente non corrisponde a quella corretta (esempio 0 o un punteggio negativo)

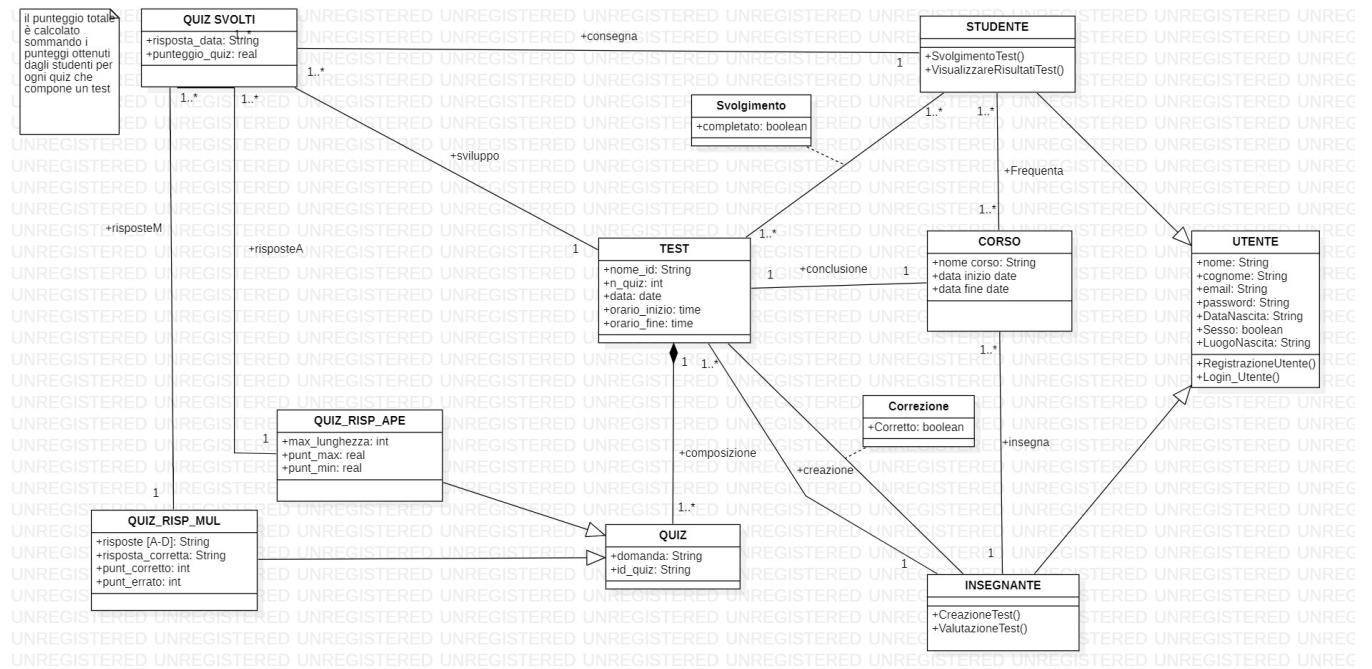
Nome	Descrizione
Quiz_svolti	Risposta_data <i>Varchar[200]</i> : Risposta data dallo studente Punteggio_quiz_dato <i>Real</i> : Il punteggio ottenuto dallo studente in base alla correttezza della risposta

Capitolo 2

PROGETTAZIONE

2.1 Diagramma del dominio del problema

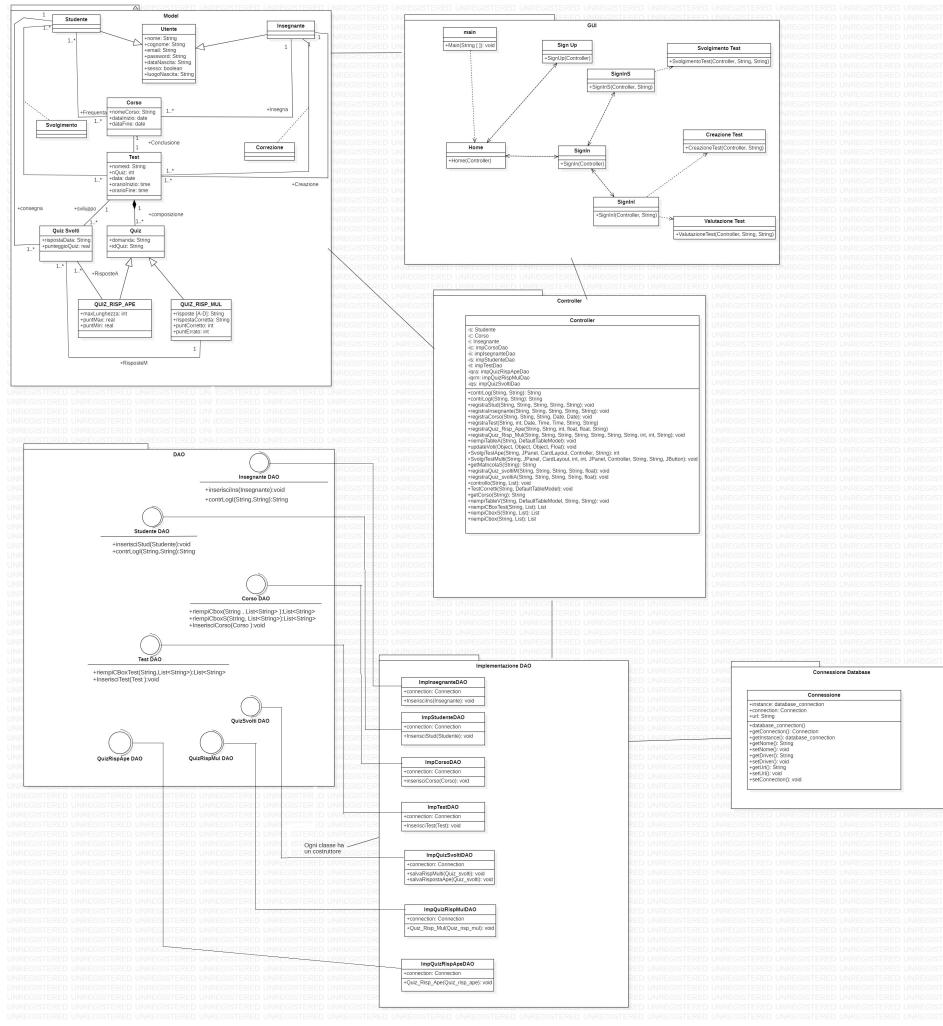
In questa sezione si mostra il diagramma delle classi contenente tutti i requisiti da noi individuati nella traccia



2.2 Diagramma del dominio della soluzione del problema

Il diagramma è costituito da una serie di package:

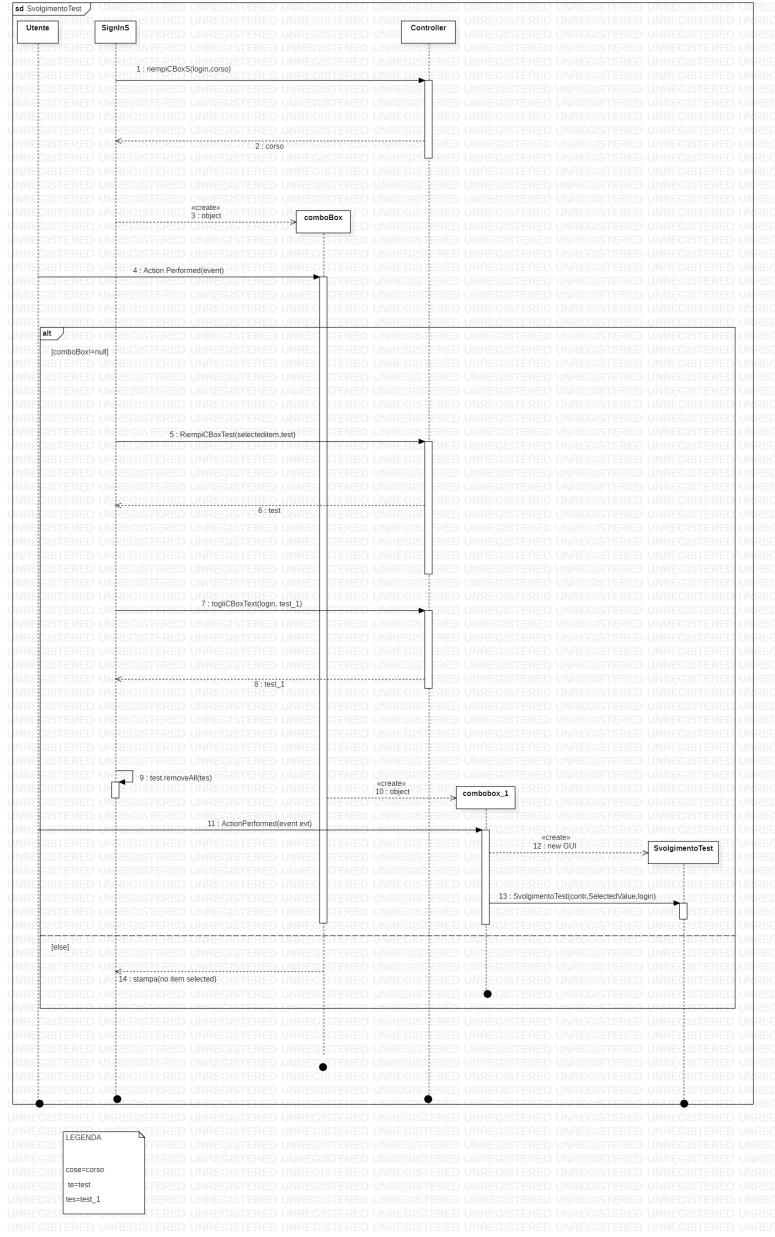
- Model: contiene tutte le classi individuate nella traccia con i rispettivi attributi e responsabilità
- GUI: contiene tutti i JFrame per l'interfaccia grafica
- DAO: contiene le interfacce per poter comunicare col database
- Implementazione DAO: ogni classe di tale package implementa le interfacce DAO per il database
- Controller: contiene la classe controller al cui interno sono istanziati gli oggetti delle classi di ImplementazioneDAO, e si potranno chiamare i loro metodi. La GUI userà il controller come tramite, poichè è l'unica classe che può comunicare con Implementazione DAO
- Connessione database: si occupa di connettere il database



2.3 Sequence diagram

2.3.1 Svolgimento Test:

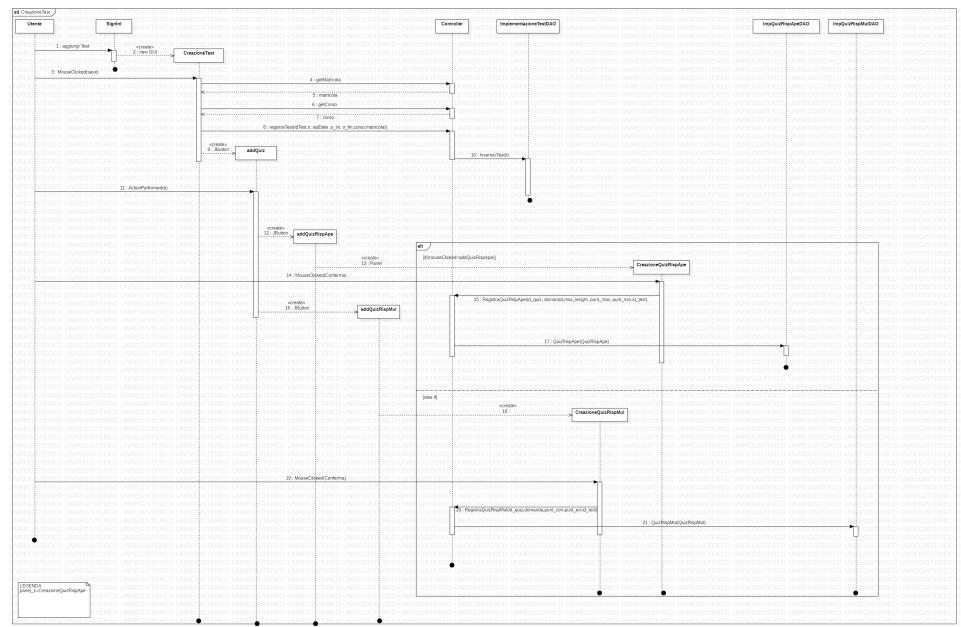
Dalla GUI SignInS viene chiamato il controller tramite la funzione `riempিCBoxS()` per riempire la comboBox con i corsi che lo studente segue. Viene poi creata una comboBox riempita dalla funzione precedente e dalla quale parte un evento che si risolve in un costrutto if/else. Nell'if la condizione è che l'elemento selezionato nella combobox sia diverso da null. In tal caso è chiamato nuovamente il controller tramite la funzione `riempিCBoxTest`, che ritorna la lista di tutti i test del corso selezionato a cui lo studente è iscritto, che poi viene successivamente svuotata dei test già svolti (in modo che lo studente non possa ripeterli). In seguito viene creata una seconda comboBox con i test del corso selezionato. Per ognuno di essi, una volta scelto, si apre una GUI che permette di svolgere il test. Se, invece l'item selezionato nella prima comboBox è null si fa una stampa.



2.3.2 Creazione Test:

L'utente(insegnante) preme il bottone che gli permette di aggiungere un test.Tale bottone apre una nuova GUI "Creazione Test" dalla quale è possibile creare il nuovo test. Dopo aver salvato le informazioni iniziali del test(es nome,data,orario) passando la funzione "registraTest" al controller che a sua volta comunica con la classe di implementazioneDAO del test. Si aggiunge il test al database e successivamente si crea il bottone che permette di aggiungere i quiz.Essi possono essere a risposta aperta o a risposta multipla, in base al tipo di quiz che si vuole inserire esiste l'apposito bottone che permette di farlo. In entrambi i casi, premendo sul bottone si apre un nuovo pannello in cui è possibile creare il quiz e poi tramite la funzione

"registra quiz", passata al controller, che comunica con l'implementazione DAO dell'apposito quiz, è possibile salvare il quiz nel database



Capitolo 3

MANUALE D'USO

Dal seguente link è possibile aprire la repository di GitHub contenente l'intero codice e i vari diagrammi : https://github.com/NoemiSpera/PROGETTO_B.D.O.O

Per poter accedere inserire le seguenti credenziali:

Insegnante

login: G.Arma

password: Gior2121@

Studente1

login: marz.pir

password: Marz2121@

Studente2

login: no.spera

password: Noem1717@