

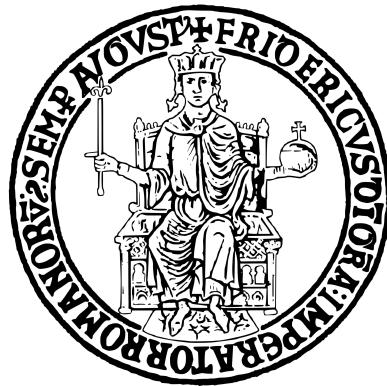
UNIVERSITÀ DEGLI STUDI DI NAPOLI  
FEDERICO II

OBJECT ORIENTATION

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

Marzia PIROZZI N86003545  
Noemi SPERA N86003717

Marzo 2022



# Indice

<b>1 DESCRIZIONE DEL PROGETTO</b>	<b>2</b>
1.1 Identificazione delle classi e delle responsabilità . . . . .	3
<b>2 PROGETTAZIONE</b>	<b>5</b>
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
<b>3 MANUALE D'USO</b>	<b>10</b>

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

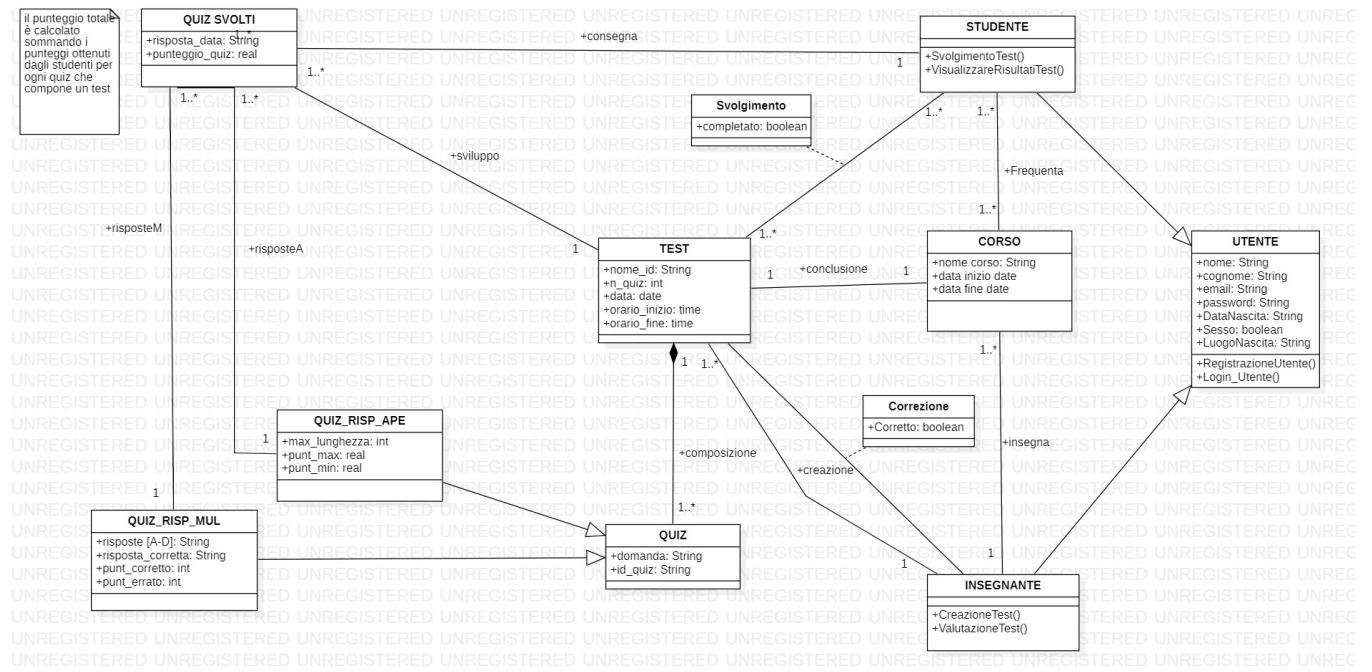
Nome	Descrizione
<b>Quiz_svolti</b>	<b>Risposta_data</b> <i>Varchar[200]</i> : Risposta data dallo studente <b>Punteggio_quiz_dato</b> <i>Real</i> : Il punteggio ottenuto dallo studente in base alla correttezza della risposta

# Capitolo 2

## PROGETTAZIONE

### 2.1 Diagramma del dominio del problema

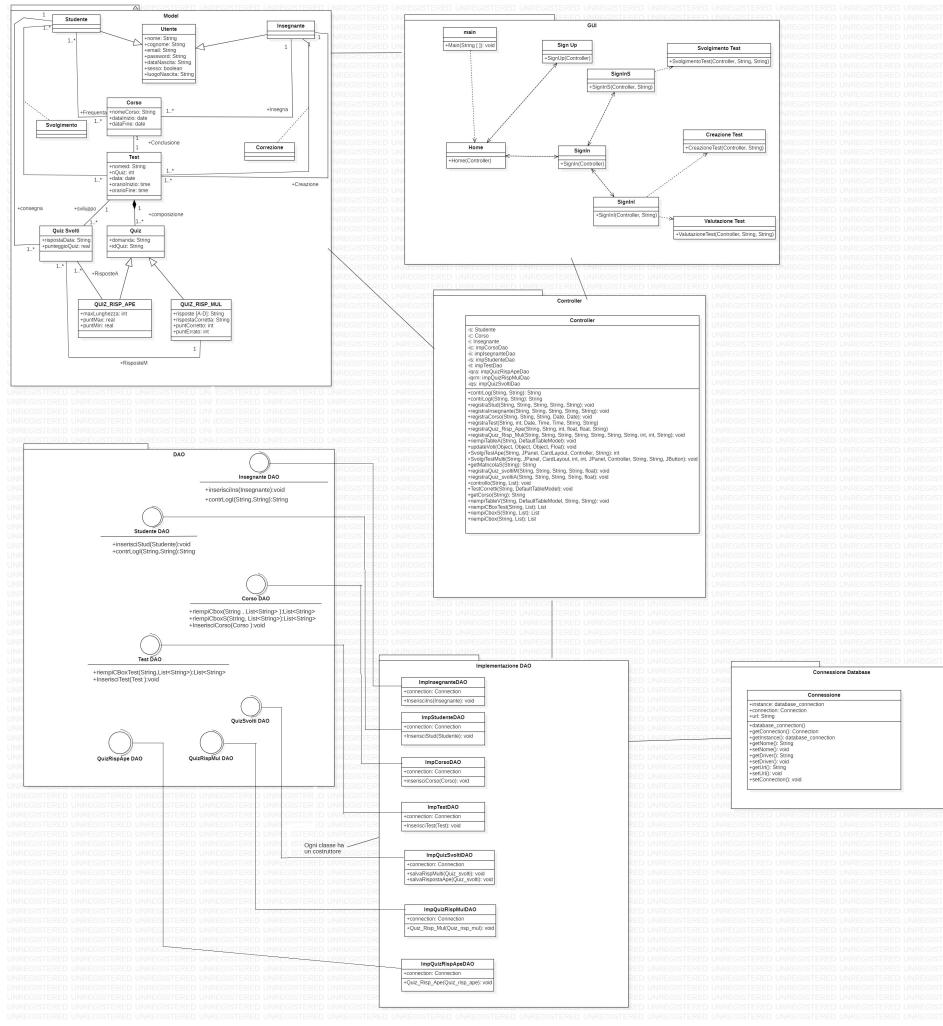
In questa sezione si mostra il diagramma delle classi contenente tutti i requisiti da noi individuati nella traccia



## 2.2 Diagramma del dominio della soluzione del problema

Il diagramma è costituito da una serie di package:

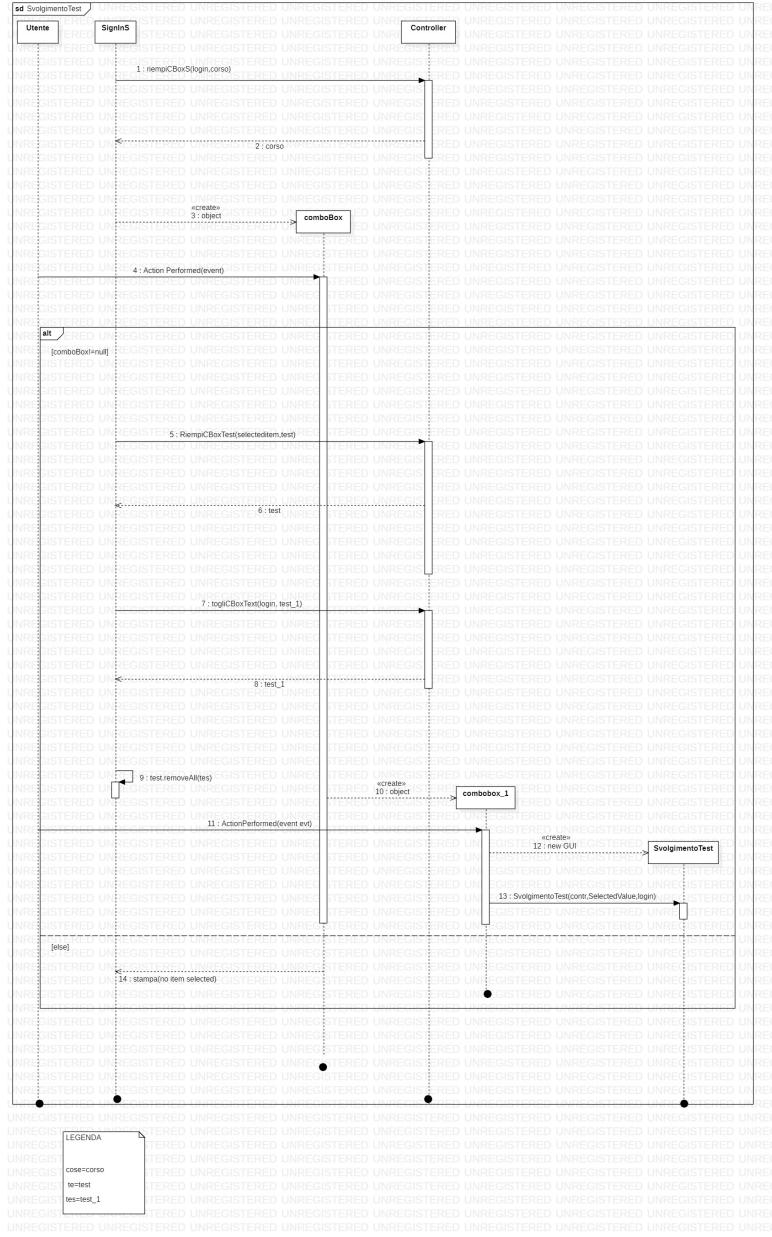
- Model: contiene tutte le classi individuate nella traccia con i rispettivi attributi e responsabilità
- GUI: contiene tutti i JFrame per l'interfaccia grafica
- DAO: contiene le interfacce per poter comunicare col database
- Implementazione DAO: ogni classe di tale package implementa le interfacce 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:

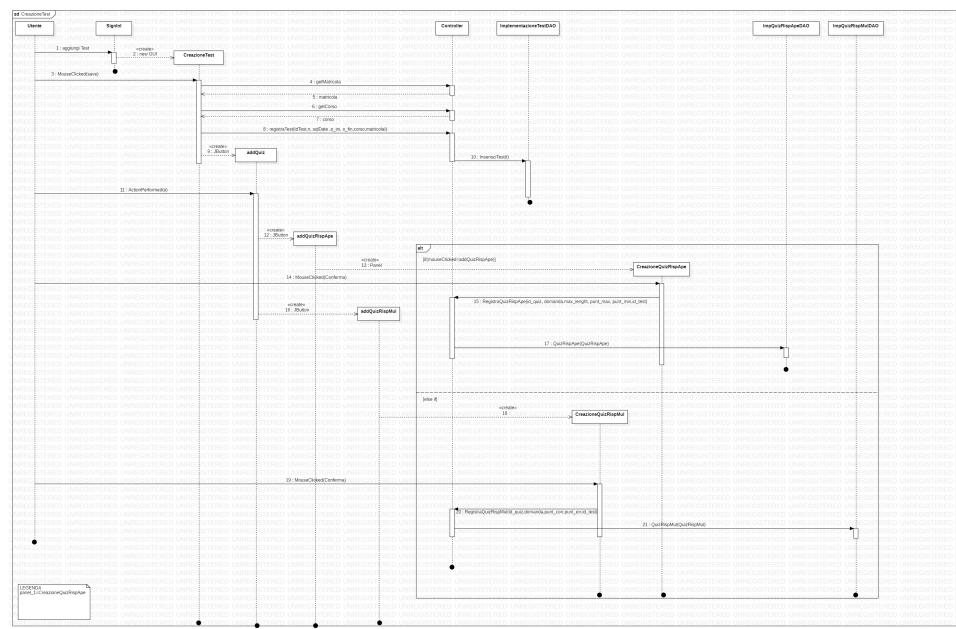
Dalla GUI SignInS viene chiamato il controller tramite la funzione riempিCBoxS() per riempire la comboBox con i corsi che lo studente segue. Viene poi creata una comboBox riempita dalla funzione precedente e dalla quale parte un evento che si risolve in un costrutto if/else. Nell'if la condizione è che 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](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@