



Applicazione di E-LEARNING tramite quiz

Marco Pastore Carmine Mascia
N86003875 N86003800
30 agosto 2022

Indice

1	Descrizione del progetto	3
1.1	Descrizione e analisi del progetto	3
1.2	Requisiti identificati	4
2	Progettazione	5
2.1	Diagramma delle classi	5
2.2	Diagramma della soluzione	7
2.3	Sequence Diagram	9
2.3.1	Login	9
2.3.2	Conferma	10
3	Manuale d'uso	11

Capitolo 1

Descrizione del progetto

1.1 Descrizione e analisi del progetto

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da 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. Lo studente può stampare l'insieme di tutti i risultati dei test che ha valutato. Lo studente può consultare il punteggio ottenuto dal suo test solo dal momento in cui l'insegnante ha completato la valutazione.

1.2 Requisiti identificati

Leggendo attentamente la traccia, abbiamo individuato le seguenti classi:

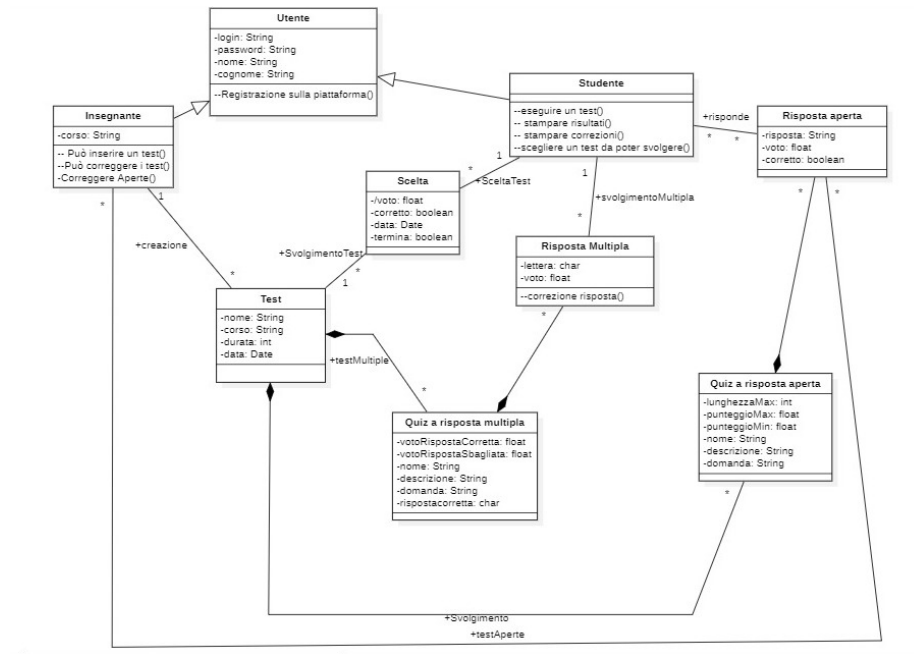
- Per gestire gli utenti abbiamo una classe **Utente** da cui ereditano:
 - **Insegnante:** un tipo di utente che ha la possibilità di creare e correggere i test affibbiando un voto.
 - **Studente:** un utente che può svolgere i test e visualizzare i risultati.
- **Scelta:** uno studente può scegliere quale test fare.
- Abbiamo una classe **Test** che racchiude:
 - **Quiz a risposta aperta:** il quiz in cui lo studente deve inserire la risposta. A sua volta include:
 - **risposta aperta:** la risposta che da lo studente.
 - **Quiz a risposta multipla:** il quiz in cui vengono fornite delle risposte di default e lo studente ne deve selezionare una. A sua volta include:
 - **risposta multipla:** la risposta che da lo studente.

Capitolo 2

Progettazione

2.1 Diagramma delle classi

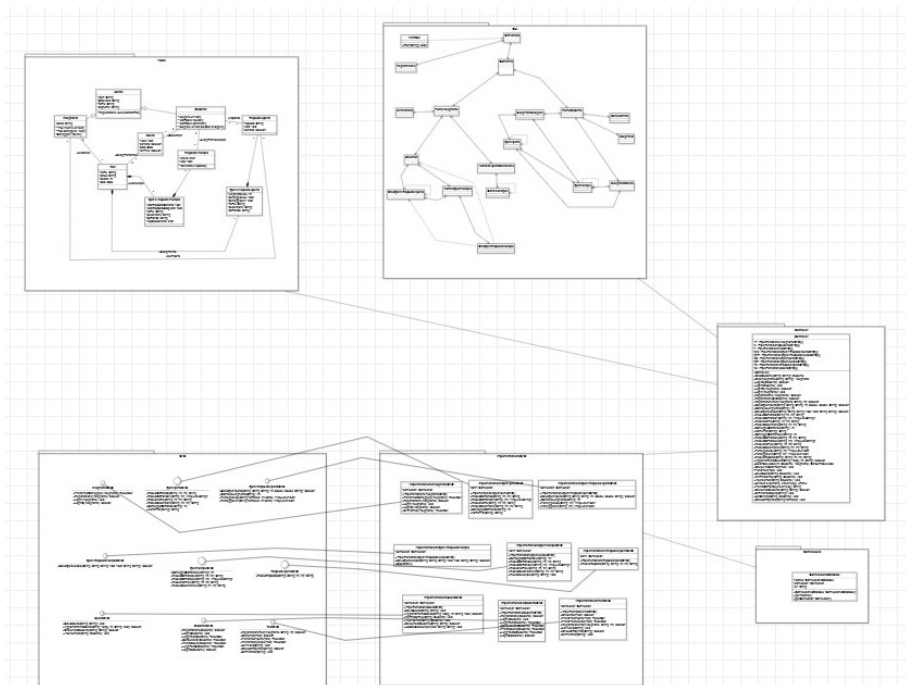
In questa sezione mostriamo il diagramma delle classi del problema in base ai requisiti che abbiamo individuato.



2.2 Diagramma della soluzione

Vediamo ora il diagramma della soluzione:

- **Model:** in questo package vi sono le classi che abbiamo individuato, sono state apportate alcune modifiche rispetto al diagramma delle classi per poterle adattare al database.
- **DAO:** qui abbiamo le interfacce per poter ricevere e inviare dati al database.
- **ImplementazioneDao:** ogni classe del package ImplementazioneDao implementa le interfacce DAO per il database postgres.
- **Connessione:** qui abbiamo la classe connessione che si occupa della comunicazione con il database.
- **Controller:** qui vi è la classe controller, al suo interno saranno istanziati gli oggetti delle classi di ImplementazioneDao, e si potranno chiamare i loro metodi. La GUI userà il controller come tramite, poiché è l'unica classe che può comunicare con le ImplementazioniDao e quindi ricevere e inviare dati al database.
- **GUI:** in questo package ci sono tutti i JFrame per l'interfaccia grafica.

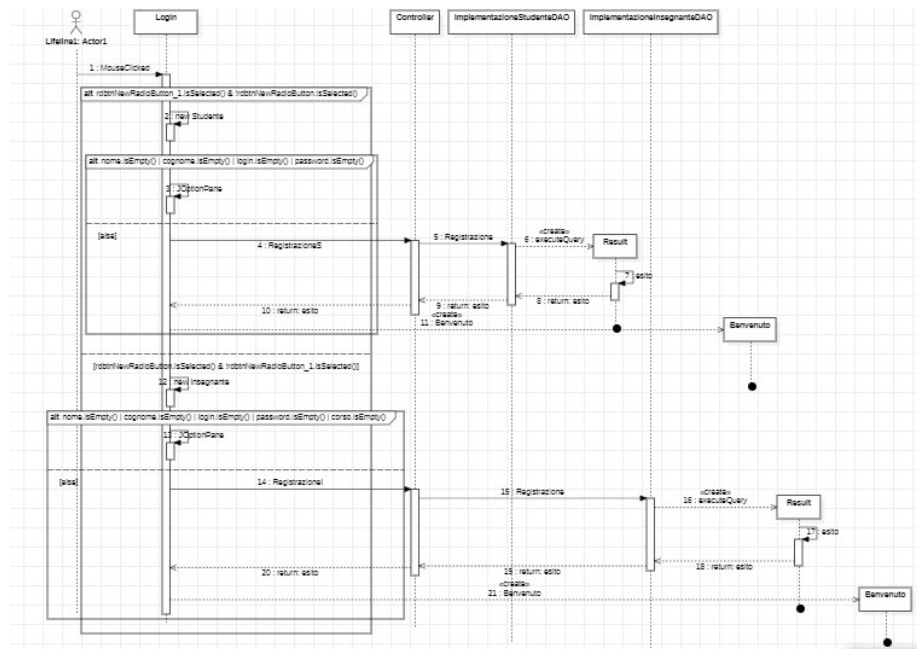


2.3 Sequence Diagram

Mostriamo il sequence diagram del login e quello dell'ascolto di una traccia.

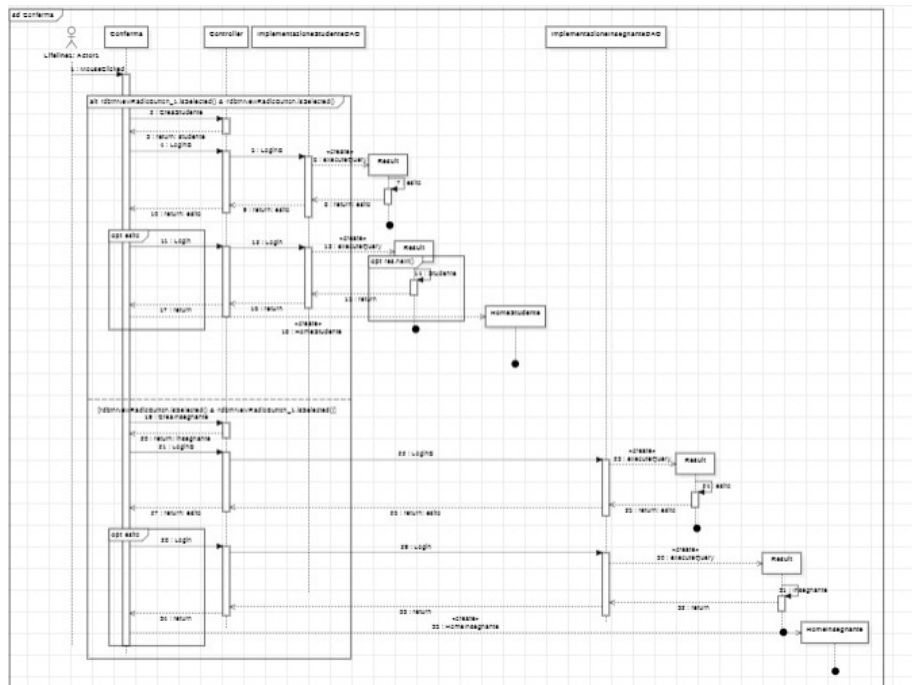
2.3.1 Login

Quando l'utente parte dalla schermata di benvenuto clicca il pulsante NO, va nella schermata di Login e inserisce i dati. Dopo aver inserito i dati e clicca il pulsante studente e non quello insegnante, allora viene allocato un nuovo studente e se non ha compilato correttamente i campi desiderati, allora apparirà un pannello di errore, altrimenti viene chiamato il controller che comunica con ImplementazioneStudenteDAO che si occupa della registrazione dello studente e ritorna alla schermata di benvenuto. Quando l'utente parte dalla schermata di benvenuto clicca il pulsante NO, va nella schermata di Login e inserisce i dati. Dopo aver inserito i dati e clicca il pulsante insegnante e non quello studente, allora viene allocato un nuovo studente e se non ha compilato correttamente i campi desiderati, allora apparirà un pannello di errore, altrimenti viene chiamato il controller che comunica con ImplementazioneStudenteDAO che si occupa della registrazione dello studente e ritorna alla schermata di benvenuto.



2.3.2 Conferma

Quando l'utente parte dalla schermata di benvenuto clicca il pulsante SI, va nella schermata di Conferma. Se l'utente inserisce login e password e clicca il bottone dello studente, si fa prima un check nel db passando prima dal Controller e poi in ImplementazioneStudenteDAO e in base al risultato di esito (se è vero), estrapola i dati passando prima dal Controller e poi in ImplementazioneStudenteDAO e lo porta nella schermata HomeStudente. Quando l'utente parte dalla schermata di benvenuto clicca il pulsante SI, va nella schermata di Conferma. Se l'utente inserisce login e password e clicca il bottone dell'insegnante, si fa prima un check nel db passando prima dal Controller e poi in ImplementazioneInsegnanteDAO e in base al risultato di esito (se è vero), estrapola i dati passando prima dal Controller e poi in ImplementazioneInsegnanteDAO e lo porta nella schermata HomeInsegnante.



Capitolo 3

Manuale d'uso

Da [questo link](#) è possibile aprire la repository di GitHub dove è possibile vedere l'intero codice e altre informazioni.

Per poter accedere come studente le consiglio le seguenti credenziali:

- **email:** Ostico@gmail.com
- **password:** 1927 E infine cliccare sul bottone Studente.

Per poter accedere come insegnante le consiglio le seguenti credenziali:

- **email:** echiaro@gmail.com
- **password:** ahokok E infine cliccare sul bottone Insegnante.

P.S. Ci sono anche molti altri account studenti e insegnanti, nel caso possiamo vedere insieme con quale altro account vuole accedere alla piattaforma. Il database se lo deve scaricare. Nel caso in cui lei voglia creare un account studente o insegnante, non c'è problema!