

***PROGETTO ISPW ANNO 2025/2026***

# ***TutorOnline***

A cura di: Del Gaudio Lorenzo

Matricola: 0308237

# 1 Software Requirement Specification

## 1.1 Introduzione

### 1.1.1 Scopo del documento

Il seguente documento intende mostrare una panoramica dettagliata del software **TutorOnline** e delle fasi di progettazione che hanno reso possibile la sua realizzazione. Il sistema è stato creato come progetto per il corso di Ingegneria del Software e Progettazione Web, svoltosi nell'anno accademico 2024/2025.

### 1.1.2 Panoramica del sistema definito

Il sistema nasce con lo scopo di fornire una piattaforma intuitiva ed efficiente per mettere in contatto studenti e tutor, facilitando l'organizzazione di lezioni private e il monitoraggio della qualità del servizio.

Gli utenti registrati come **Studenti** avranno la possibilità di cercare tutor filtrando per le materie di interesse, richiedere appuntamenti in base alle disponibilità e, una volta conclusa la lezione, inviare recensioni. Una particolarità del sistema è l'integrazione di un motore di intelligenza artificiale per l'analisi del *sentiment* delle recensioni, che garantisce una valutazione oggettiva del feedback lasciato.

Gli utenti registrati come **Tutor** potranno gestire le richieste di appuntamento in entrata (accettandole o rifiutandole), confermare l'avvenuto svolgimento delle lezioni e consultare le recensioni ricevute dagli studenti, monitorando così la propria reputazione sulla piattaforma.

### 1.1.3 Requisiti HW e SW

#### Software utilizzato per lo sviluppo:

- **IntelliJ IDEA** come ambiente di sviluppo integrato (IDE).
- **Draw.io** per la modellazione UML e la creazione dei diagrammi di progettazione.
- **Scene Builder** per la progettazione delle interfacce grafiche (file FXML).
- **MySQL** per la gestione del database relazionale.
- **Python** per l'esecuzione del modulo di analisi del sentiment.
- **Hugging Face API** (modello *Bert Uncased Sentiment*) per l'elaborazione del linguaggio naturale nelle recensioni.

**Requisiti Hardware e Sistema Operativo:** Poiché l'applicazione è basata su Java (JavaFX), è progettata per essere multiplatforma; tuttavia, i requisiti raccomandati per un'esperienza fluida, specialmente considerando l'interazione con il modulo Python locale, sono i seguenti:

- **Sistemi operativi:**
  - Windows 10 / 11 (Ambiente di sviluppo e test principale).
  - MacOS v10.14 o superiore.
  - Linux (distribuzioni recenti con supporto JDK 21+).
- **Processore:** Consigliato Intel Core i5 / AMD Ryzen 5 o superiore (per gestire database e server locale simultaneamente).

- **Memoria (RAM):** Minimo 4 GB, consigliato 8 GB o superiore.
- **Hard Disk:** Spazio minimo richiesto 2 GB (inclusi JDK e dipendenze Python).
- **Connessione Internet:** Necessaria per l'interrogazione delle API esterne (Hugging Face) e, nel caso di configurazione remota, per l'accesso al Database.

#### 1.1.4 Sistemi Correlati

Per analizzare il contesto di dominio, sono stati presi in esame i seguenti sistemi esistenti:

- **Superprof**
  - **PRO:** Database globale molto vasto, permette di trovare tutor per qualsiasi disciplina, anche non scolastica.
  - **CONTRO:** Spesso richiede un pagamento ("Pass Alunno") solo per poter contattare i tutor, creando una barriera all'ingresso per l'utente occasionale.
- **GoStudent**
  - **PRO:** Offre percorsi di apprendimento strutturati e tutor verificati con colloquio.
  - **CONTRO:** Modello di business basato su abbonamenti a lungo termine e pacchetti di lezioni costosi, che riducono la flessibilità per chi cerca un aiuto saltuario.

## 1.2 User Stories

1. Come **Studiante**, voglio ricercare i tutor filtrandoli per materia, in modo da visualizzare e scorrere solo i profili che soddisfano le mie esigenze di studio.
2. Come **Studiante**, voglio richiedere la prenotazione di un appuntamento con un tutor specifico, in modo da fissare una lezione in una data e orario disponibili.
3. Come **Tutor**, voglio gestire le richieste di prenotazione ricevute (accettandole o rifiutandole), in modo da aggiornare lo stato dell'appuntamento e organizzare la mia agenda.

### Dizionario

- **Stato della prenotazione:** In attesa, Accettata, Rifiutata, Completata.

Hai ragione, "deve permettere" è spesso considerato un requisito utente mascherato. I requisiti funzionali descrivono **l'azione attiva del sistema**.

Ecco la versione corretta, formale e diretta, pronta per il copia-incolla:

---

## 1.3 Functional Requirements

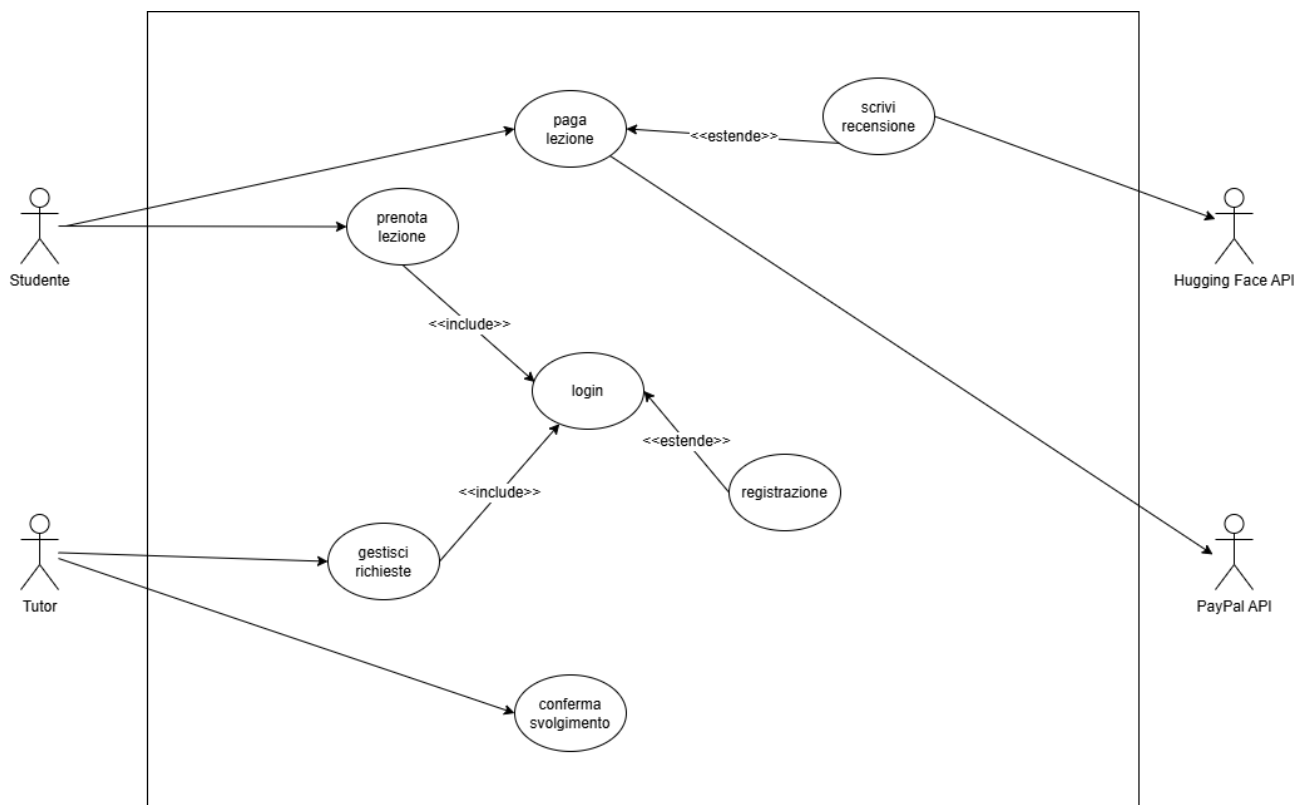
1. Il sistema deve **selezionare** i profili dei tutor che corrispondono alla materia di interesse indicata dallo studente.
2. Il sistema deve **aggiornare** lo stato della prenotazione in seguito all'accettazione, rifiuto o conclusione della lezione da parte del tutor.
3. Il sistema deve **calcolare** automaticamente il valore del sentiment associato al testo di ogni recensione inviata, assegnando un punteggio numerico.

## Dizionario:

- **Selezionare:** Il sistema filtra il database degli utenti con ruolo "Tutor" e restituisce solo quelli associati alla materia cercata.
- **Aggiornare:** Il sistema esegue la transizione di stato della richiesta (da *In attesa* a *Accettata/Rifiutata*, o da *Accettata* a *Completata*).
- **Calcolare:** Il sistema elabora il testo tramite un algoritmo di analisi del linguaggio (Sentiment Analysis) per generare un voto da 1 a 5.

## 1.4 Use Cases

### 1.4.1 Overview Diagram



### 1.4.2 Internal Steps

#### Salda e Recensisci Lezione

*\*include Login*

1. L'utente seleziona l'opzione: "Lezioni Completate".
2. Il sistema mostra la lista delle lezioni svolte, sia in attesa di pagamento sia già saldate.
3. L'utente seleziona la lezione da saldare.
4. Il sistema mostra il modulo di pagamento con i dettagli della lezione.
5. L'utente conferma la transazione.

6. Il sistema elabora il pagamento tramite il gateway esterno (PayPal).
7. Il sistema verifica l'esito della transazione.
8. Il sistema registra il pagamento e aggiorna lo stato della lezione.
9. Il sistema notifica il successo e mostra il pulsante per la recensione.
10. L'utente seleziona l'opzione di recensione.
11. Il sistema mostra il modulo della recensione.
12. L'utente compila e conferma i dati del modulo.
13. Il sistema calcola il sentiment del testo inserito.
14. Il sistema salva la recensione e il relativo punteggio.
15. Il sistema notifica all'utente l'avvenuta pubblicazione.

### Extensions:

- **6a. Il gateway di pagamento non risponde o rifiuta la transazione:** il sistema notifica il fallimento del pagamento e termina il caso d'uso mantenendo la lezione come "Da Pagare".
- **13a. L'API di Sentiment Analysis non è raggiungibile:** il sistema salva comunque la recensione assegnando un punteggio neutro di default e notifica all'amministratore un warning nei log, completando il caso d'uso per l'utente.
- **14a. Il database non risponde durante il salvataggio:** il sistema notifica un errore tecnico e invita l'utente a riprovare l'inserimento della recensione più tardi.

### Dizionario

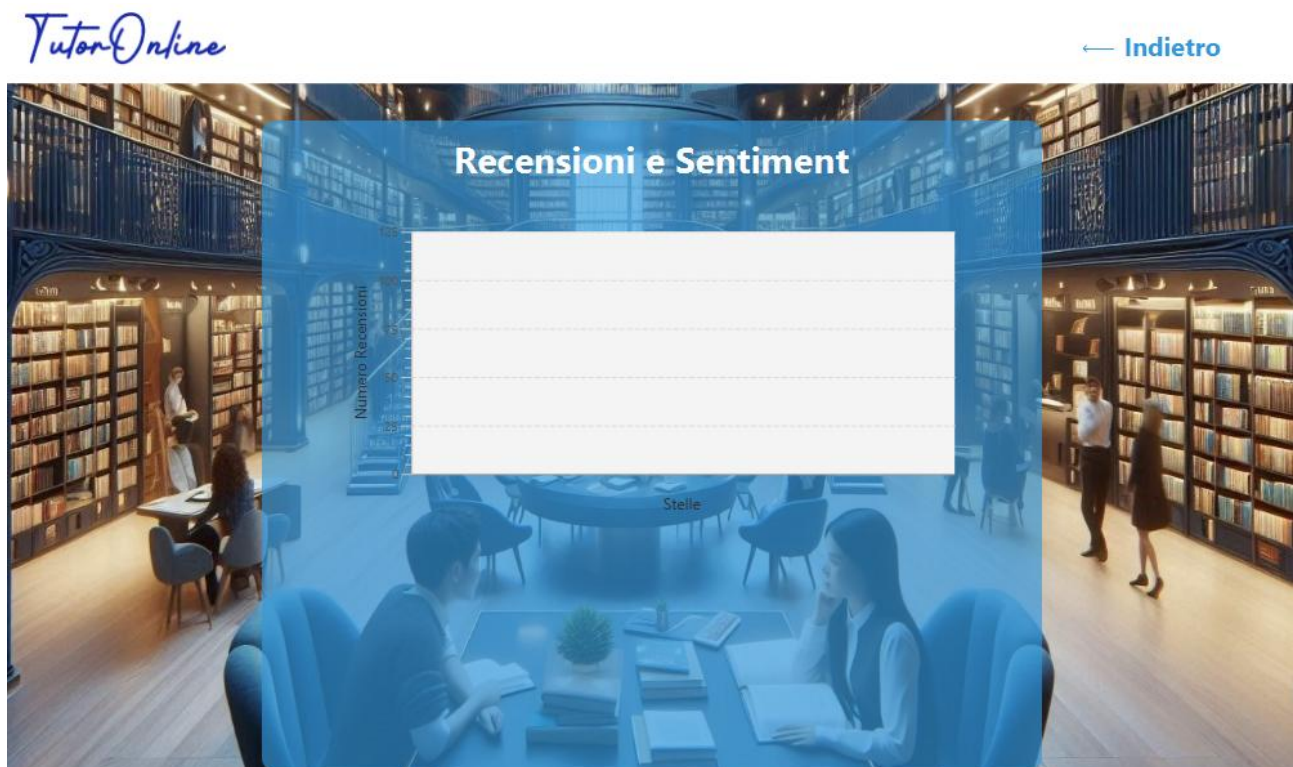
- **Lezioni Completate:** Lista delle prenotazioni per le quali il Tutor ha confermato l'avvenuto svolgimento.
- **Modulo di pagamento:** Riepilogo contenente Materia, Data, Nome Tutor.
- **Modulo della recensione:** Campo di testo libero per il commento e visualizzazione del Tutor valutato.
- **Calcola il sentiment:** Il sistema interroga l'API esterna (Hugging Face) per attribuire un valore numerico al testo della recensione.

## 2 Storyboards

Schermata per la ricerca dei tutor per materia (Studente):



Schermata per la visualizzazione delle recensioni (Tutor):

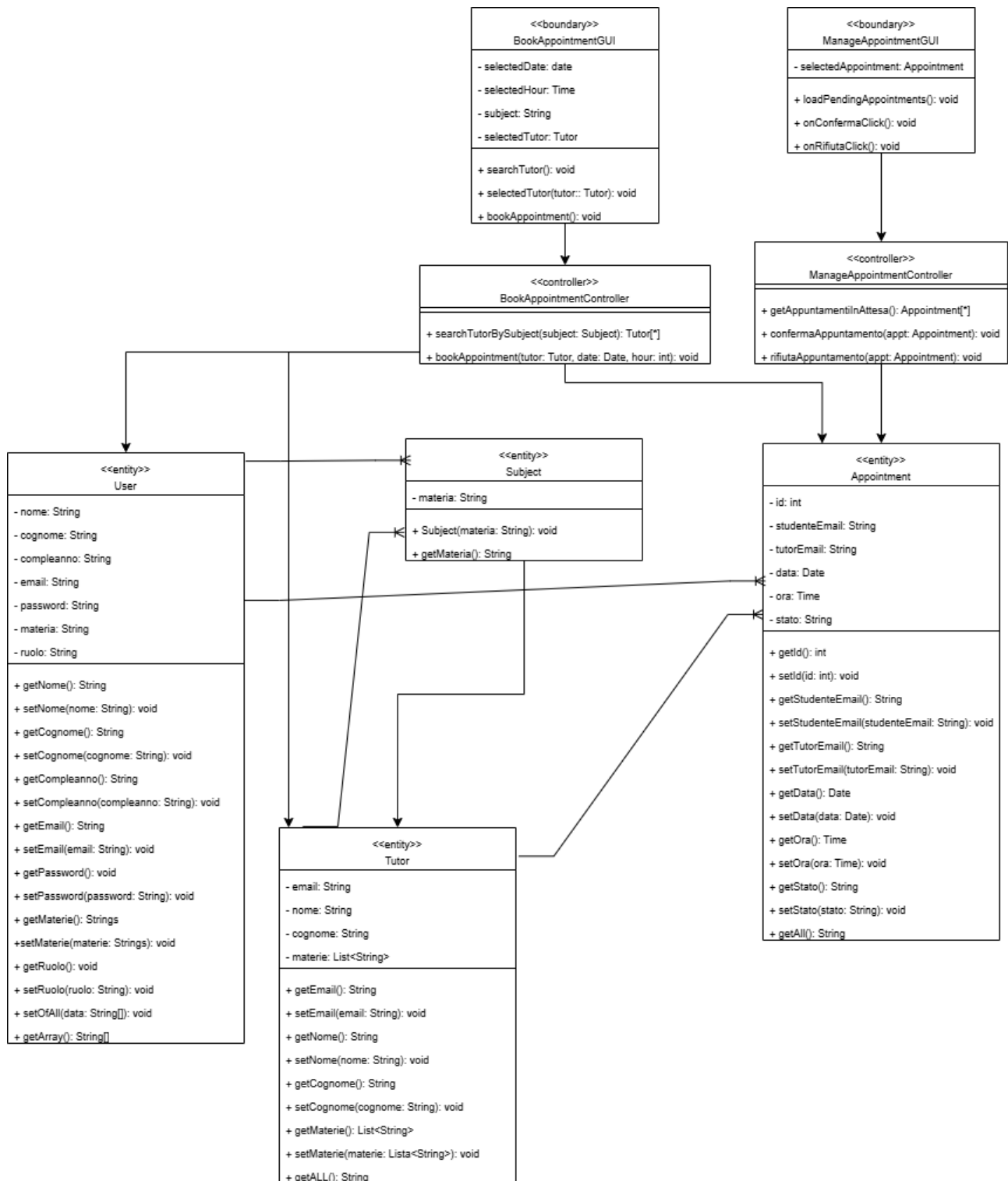


# 3 Design

## 3.1 Class Diagram

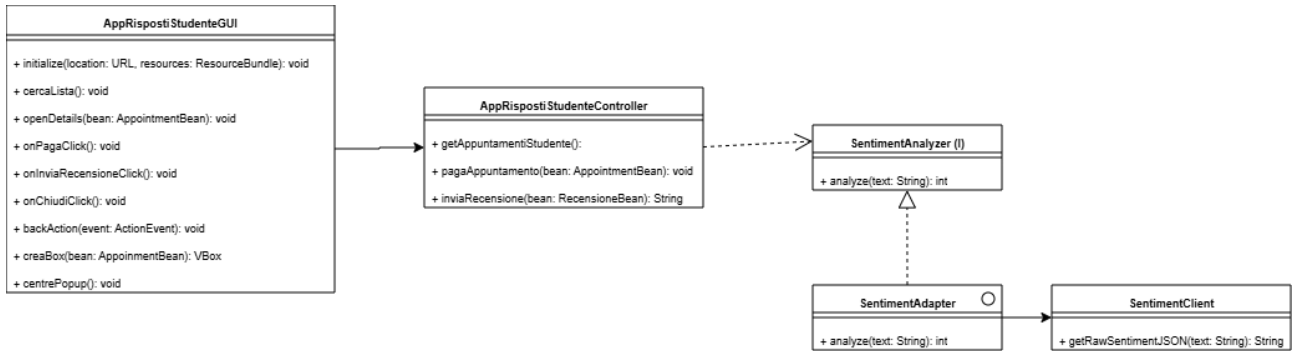
### 3.1.1 VOPC (Analysis)

- VOPC BCE relativo ai casi d'uso “Prenota lezione” e “Gestisci richieste”



### 3.1.1.1 Design-Level Diagram

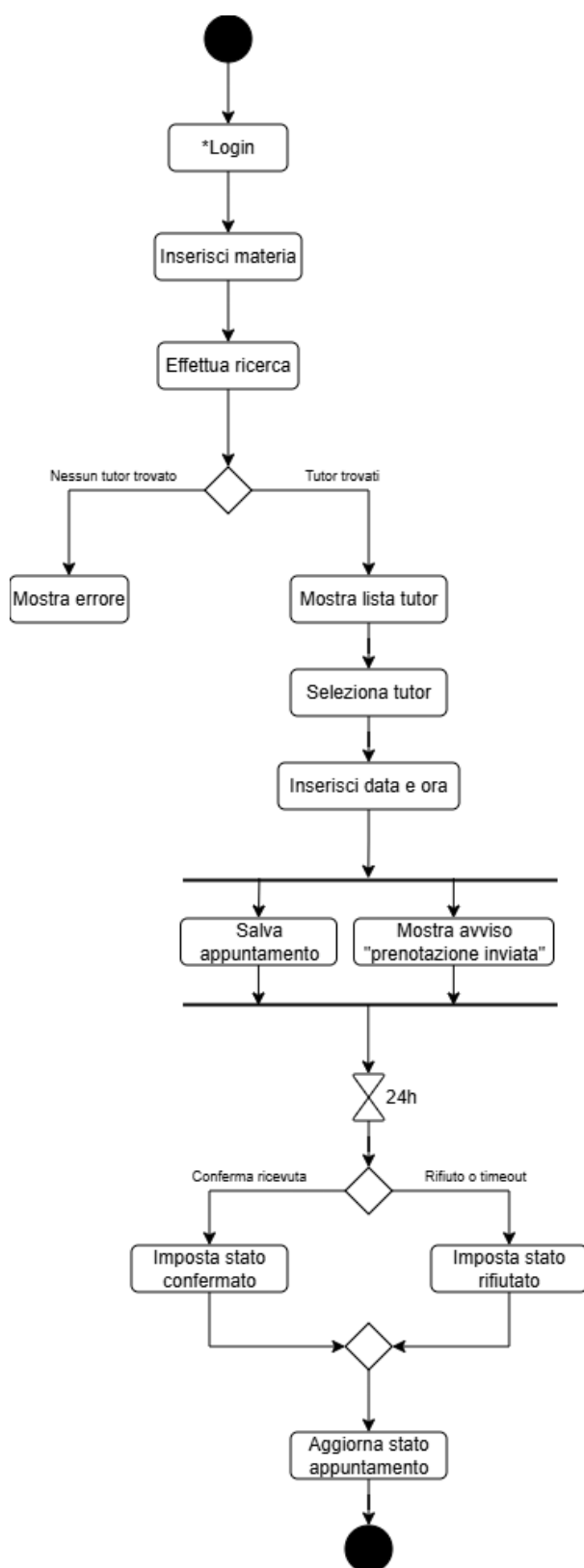
Il Design-Level riportato è relativo ad un Pattern Adapter.





## 3.2 Activity Diagram

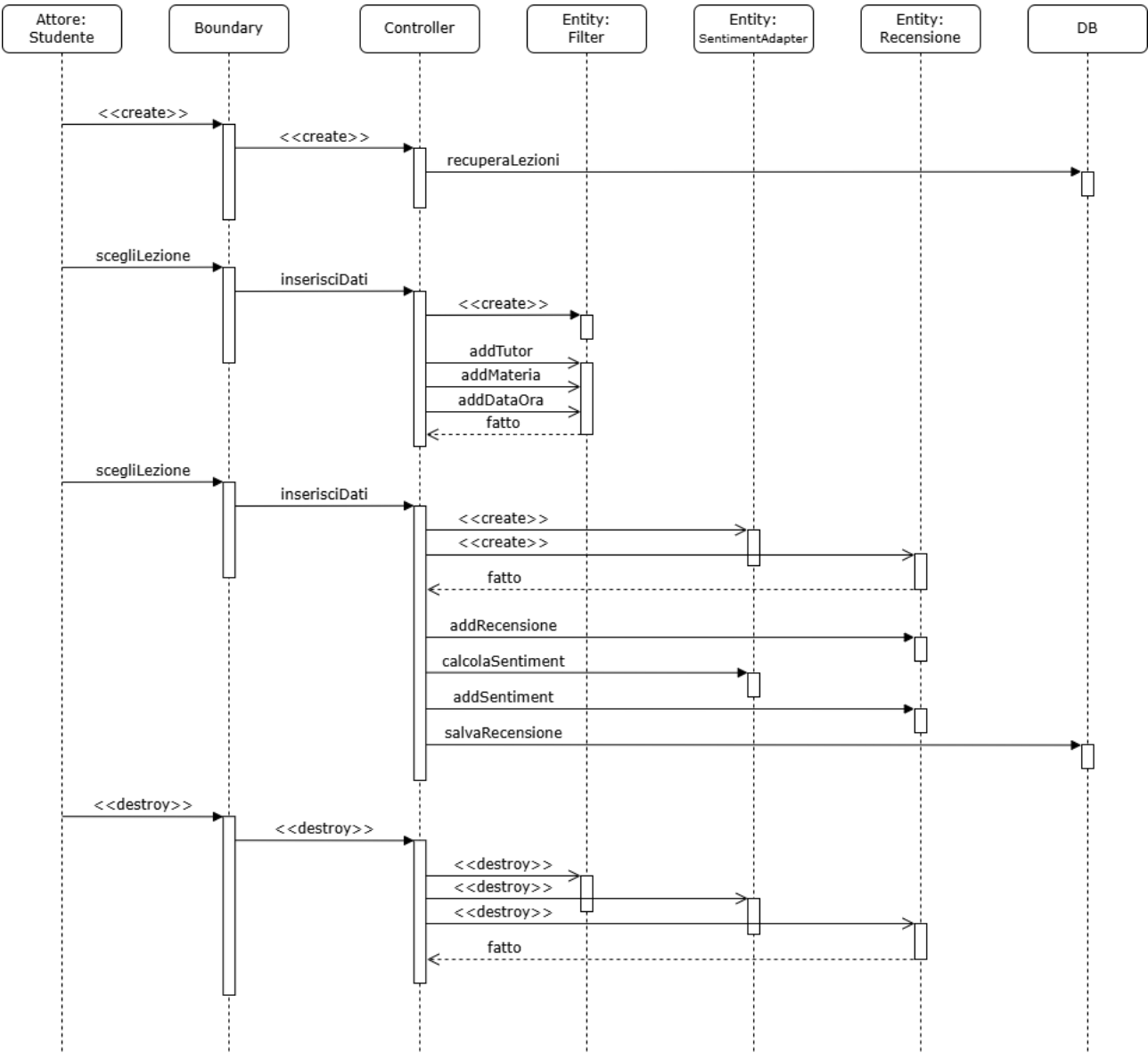
Activity Diagram relativo al caso d'uso: Prenota lezione.



**Nota:** il timer per la scadenza automatica della prenotazione (Timeout) presente nel diagramma non è stato implementato nel prototipo software.

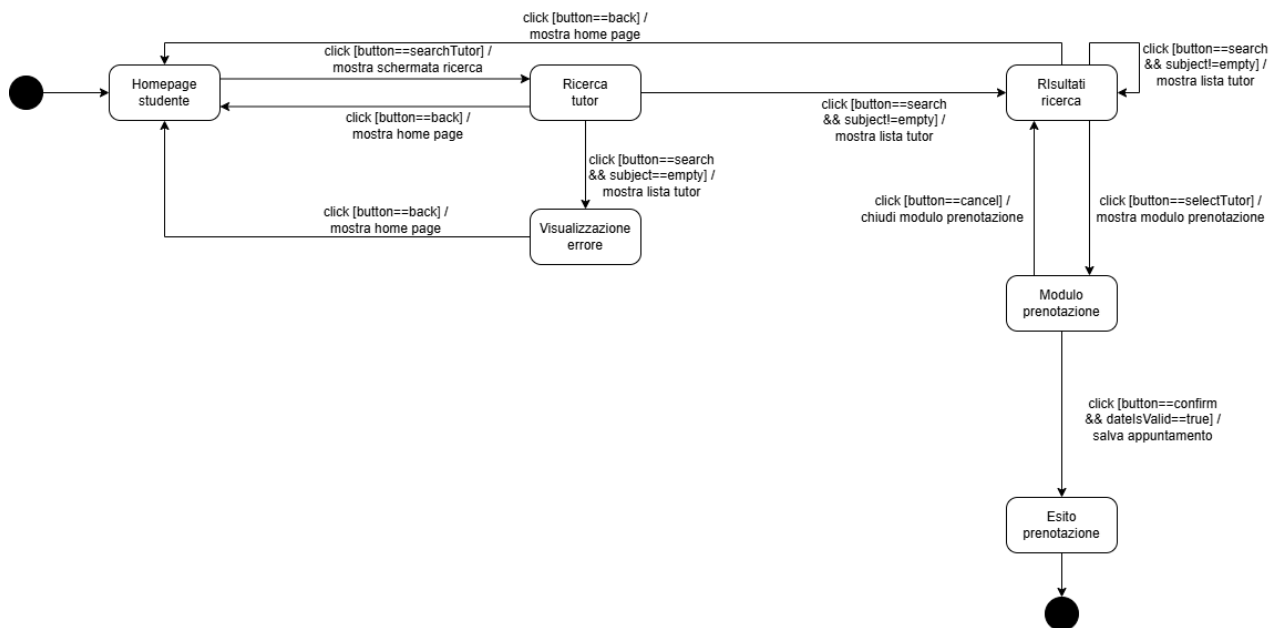
# 3.3 Sequence Diagram

Sequence Diagram relativo al caso d'uso: Scrivi recensione.



## 3.4 State Diagram

State Diagram relativo al caso d'uso: Prenota lezione.



## 4 Testing

1. Verifica che venga sollevata l'eccezione adeguata (`InvalidEmailException`) se si tenta di effettuare il login inserendo un indirizzo email dal formato sintatticamente errato (es. privo di chiocciola o dominio).
2. Verifica che non venga ammesso il tentativo di login se i campi obbligatori (email o password) vengono lasciati vuoti o sono nulli.
3. Verifica che venga sollevata l'eccezione adeguata (`UserDoesNotExistException`) se si tenta di accedere con le credenziali di un utente non registrato nel sistema.

## 5 SonarCloud

Il link di SonarCloud associato al progetto è:

[https://sonarcloud.io/project/overview?id=lollodelga\\_ISPW\\_repo](https://sonarcloud.io/project/overview?id=lollodelga_ISPW_repo)