qrLectures

2020/2021

1. Abstract

qrLectures è una piattaforma rivolta agli istituti didattici per semplificare, minimizzando il rischio di inganni, il modo in cui vengono gestite le presenze nelle lezioni. Il sistema si basa su due codici qr che, per ogni lezione, verranno mostrati agli studenti in aula, rispettivamente all'inizio e alla fine della lezione. Ogni qualvolta uno studente effettua lo scan dei due codici, relativi a una specifica lezione, otterrà lo status partecipante.

2. Glossario

2.1. Codice QR

Codice a barre bidimensionale impiegato, in genere, per memorizzare informazioni destinate a essere lette tramite uno smartphone.

2.2. Pseudo-partecipante

Studente che ha scannerizzato il codice QR relativo all'inizio della lezione.

2.3. Status partecipante

Studente che ha scannerizzato sia il codice QR relativo all'inizio della lezione sia quello relativo alla fine.

2.4. Unique Device Identifier (UDI)

Identificatore univoco associato a ciascun dispositivo android.

2.5. REST APIs

Application Programming Interface che rispettano l'architettura software Representational State transfer, comunemente utilizzate per creare applicazioni interattive che usano servizi Web.

2.6. JavaScript Object Notation (JSON)

Formato adatto all'interscambio di dati fra applicazioni client/server basato sugli oggetti di JavaScript.

2.7. Django

Web framework, open-source, scritto in Python, per lo sviluppo di applicazioni web.

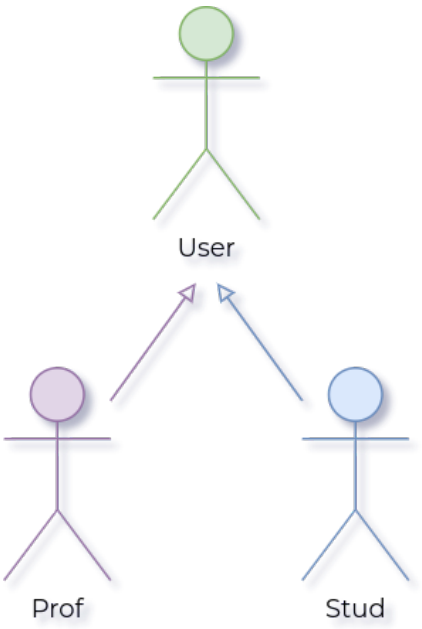
3. Analisi dei Requisiti

L'obiettivo è quello di creare un sistema **manageriale** con lo scopo di gestire in maniera, semplice, lo status di partecipazione, degli studenti, riferito alle lezioni attraverso un **protocollo** basato su codici QR. Questo protocollo, oltre a fornire immediatezza e semplicità, dovrà essere **non aggirabile**, i.e. gli studenti che risulteranno partecipanti di una, determinata, lezione, dovranno essere gli stessi che, fisicamente, hanno assistito alla lezione.

3.1. Attori del sistema

Gli utenti che andranno a interagire con il sistema sono due:

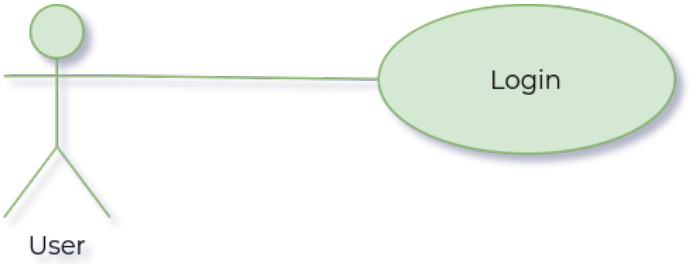
- **Professore (*prof*)**: interagirà con la piattaforma tramite web browser.
- **Studente (*stud*):** interagirà con la piattaforma tramite un'applicazione Android e avrà associato un Unique Device Identifier (UDI), cioè un identificatore che gli permetterà l'accesso al sistema, esclusivamente, da un solo dispositivo.



3.2. Requisiti

3.2.1. Autenticazione

Gli attori del sistema dovranno essere in grado di effettuare l'autenticazione nella piattaforma che gli permetterà di ottenere le autorizzazioni, necessarie, per interagirci.

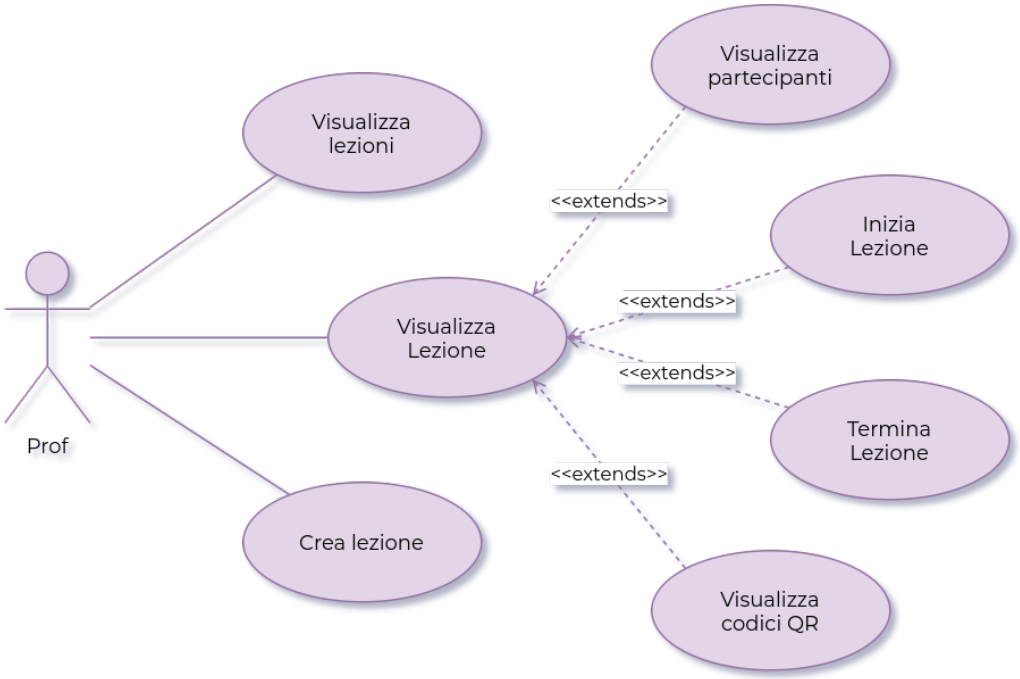


L'attore *stud*, nel processo di autenticazione, oltre a fornire le credenziali d'accesso, fornirà, automaticamente, l'**UDI** del device con il quale sta interagendo con il sistema.

3.2.2. Gestione Lezioni

Tutti gli attori saranno in grado di interagire con le lezioni in modo opportuno, rispetto al loro ruolo.
L'attore *prof* dovrà essere in grado di:

- **visualizzare un elenco di tutte le lezioni**;
- **visualizza e gestire una Lezione:**
  - *visualizzare i partecipanti (stud),*
  - *iniziare la Lezione,*
  - *terminare la Lezione,*
  - *visualizzare i codici QR rispettivi all'inizio e alla fine della Lezione;*
- **creare lezioni.**



L'attore ***stud*** dovrà, invece, essere in grado di:

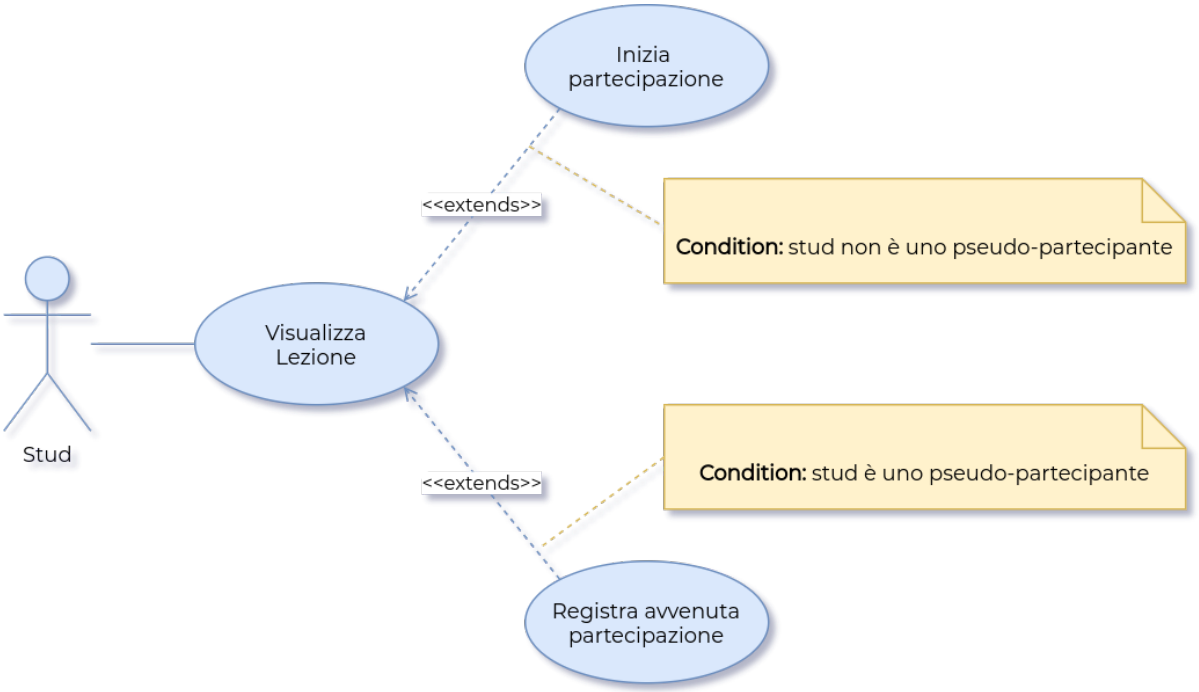
- **visualizzare tutte le lezioni in modo da poter, facilmente, distinguere:**
  - *lezioni dove stud risulta un partecipante,*
  - *lezioni dove stud risulta uno pseudo-partecipante,*
  - *lezioni prossime.*

3.2.3. Partecipazione Lezioni

L'attore *stud*, attraverso la scansione di due codici QR, dovrà essere in grado di poter ottenere lo status di partecipante per una, determinata, lezione.

Ogni qualvolta *stud* scansionerà un codice QR, relativo a Lezione, si distinguono due casi:

- *stud* **scansiona il QR relativo all'inizio** della Lezione:
    - *stud* diventa uno pseudo-partecipante della Lezione;
- *stud* **scansiona il QR relativo alla fine** della Lezione:
    - se *stud* è uno pseudo-partecipante della Lezione, *stud* acquisisce lo status partecipante,
    - se *stud* non è uno pseudo-partecipante della Lezione, *stud* non acquisisce lo status partecipante.



3.2.4. Sicurezza

Requisito, non funzionale, di vitale importanza è la sicurezza durante l'**acquisizione dello** status partecipante. Sono due i meccanismi che andranno a **minimizzare il rischio** che *stud* possa ottenere lo status partecipante, senza averne i requisiti. Essi sono i seguenti:

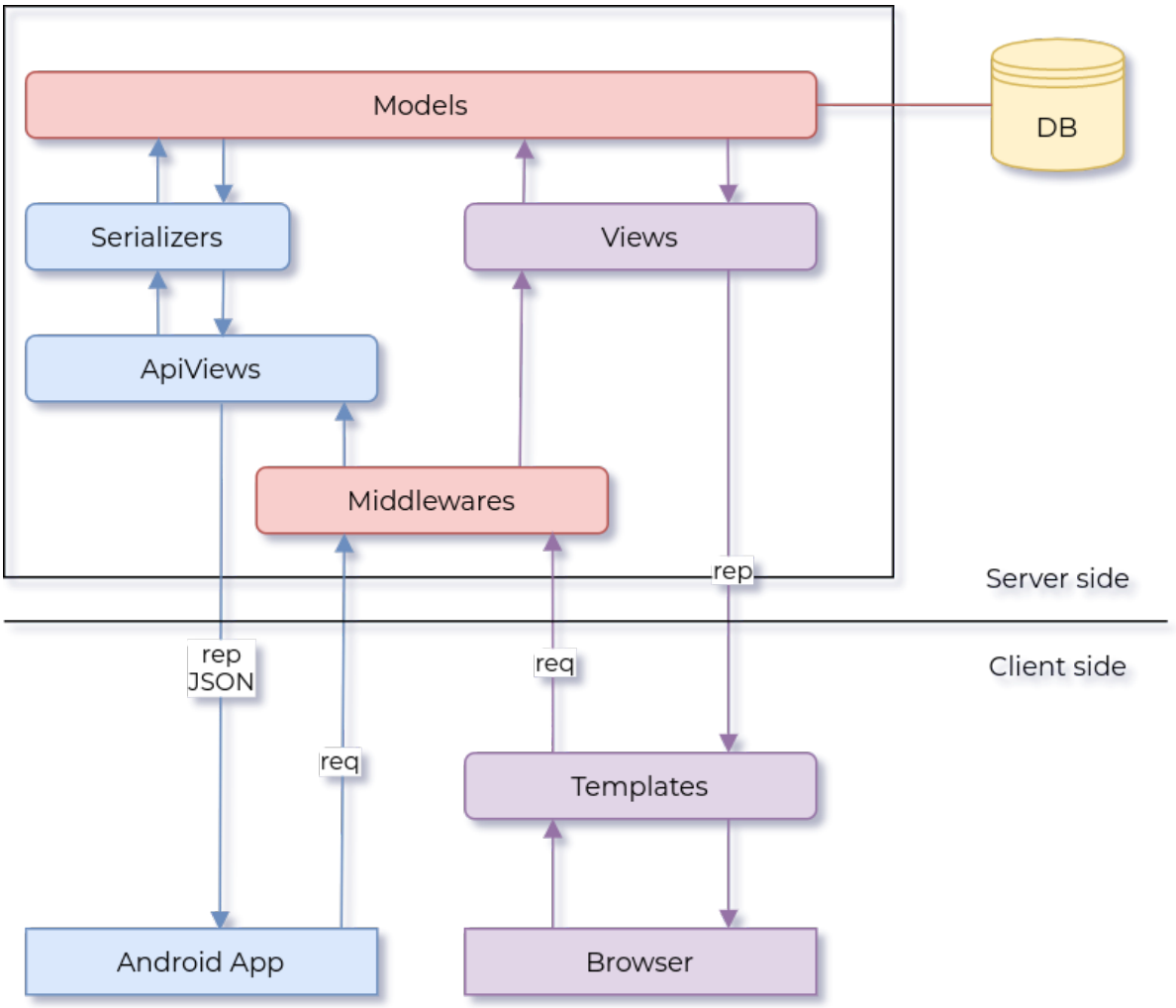
- **modifica dinamica codice QR**: il sistema aggiornerà ogni x secondi il codice QR, per evitare che possa essere, facilmente, trasmesso a *stud* non presenti in struttura, un codice QR valido;
- ogni ***stud* ha un dispositivo Android associato**, che farà in modo che *stud* potrà effettuare l'autenticazione alla piattaforma, esclusivamente, con un dispositivo. Quindi si andrà **minimizzare**, anche, il rischio che un attore, diverso da *stud*, possa fargli ottenere lo status partecipante senza averne i requisiti.

4. Architettura del sistema

La piattaforma non è altro che un sistema distribuito composto da un **web server** e un'**applicazione mobile.** Gli utenti, dipendentemente dal loro ruolo, si interfacceranno al sistema come segue:

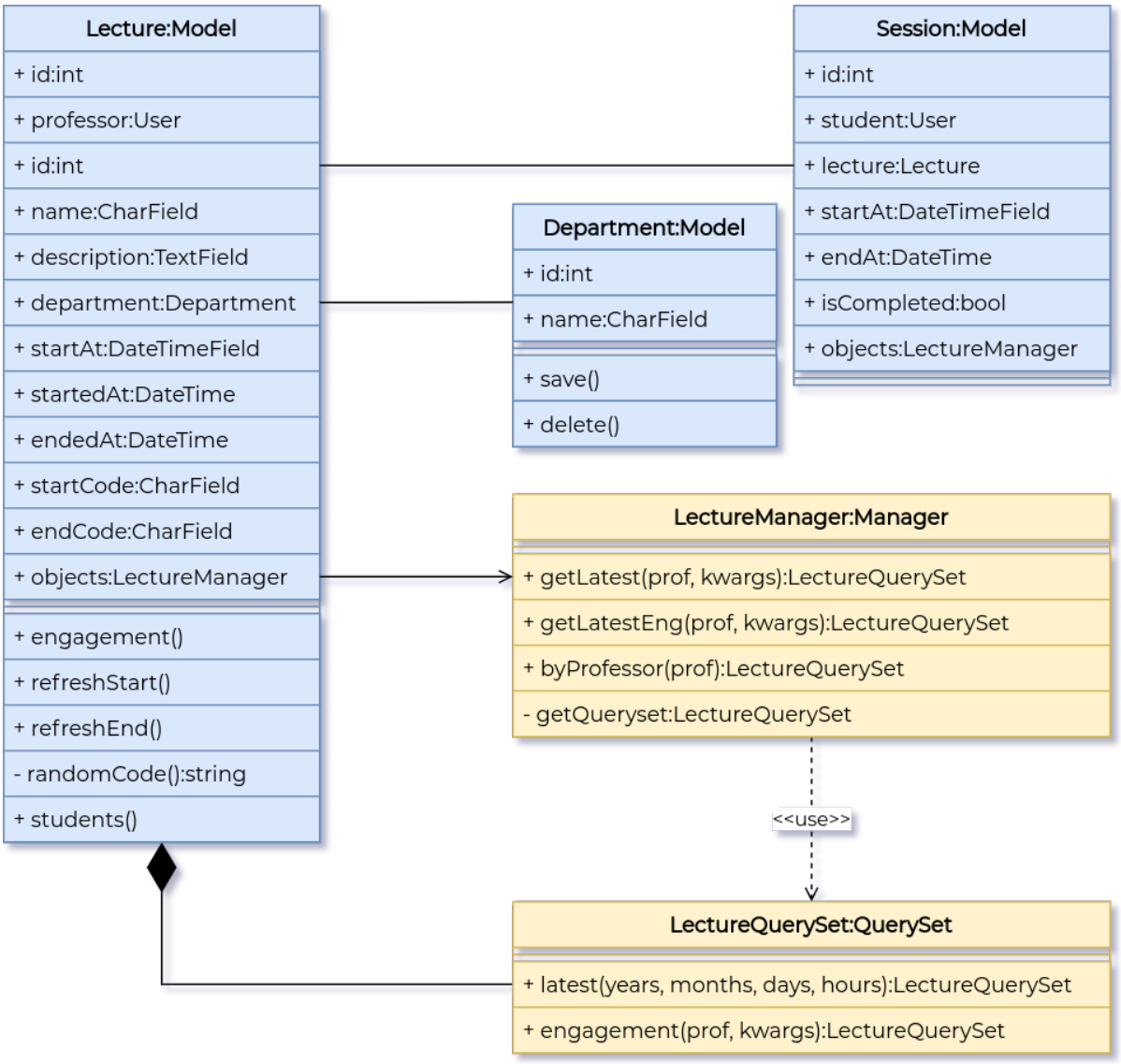
- **professori**: tramite un web browser potranno fare richieste al server e ricevere della **pagine HTML** dinamiche,
- **studenti**: con un'**applicazione Android** saranno in grado di sfruttare le REST APIs per interrogare il server, ricevendo risposte sotto forma di JSON.

L'architettura del sistema è strettamente legata alle tecnologie utilizzate per l'implementazione. I framework Django e Django Rest Framework, utilizzati per lo sviluppo dell'applicativo server e per la creazione delle pagine HTML dinamiche, si basano su una architettura MVT e su una serie di astrazioni, sotto forma di classi, che semplificano sia la progettazione che la realizzazione del sistema.



4.1. Modelli (database)

La classe ***Model*** insieme a ***Manager*** e ***QuerySet*** inseriscono un livello di astrazione tra l'applicativo e il database, permettendo l'interazione, con quest'ultimo, senza ricorrere all'uso di linguaggi di query.
Tutte le classi figlie di ***Model***, altro non sarebbero che una rappresentazione degli oggetti che si troveranno nel database. Ogni classe che eredita da *Model* ha un attributo, ***objects***: un riferimento a un'istanza di ***Manager*** che, insieme a l'oggetto ***QuerySet***, permettono di interrogare il database.



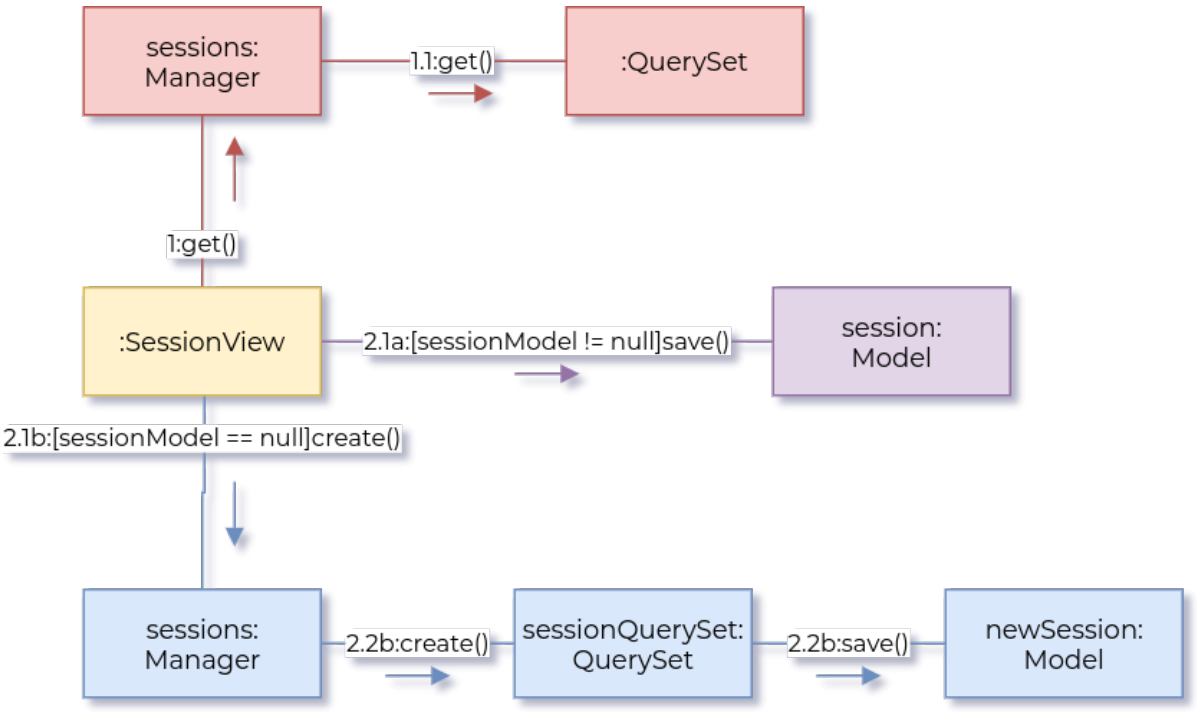
L'oggetto ***Manager*** fa da **façade** all'oggetto ***QuerySet*** permettendo la creazione di interfacce, semplici, per interrogare e gestire il database.

4.2. Partecipazione Lezione

Per partecipare a una lezione, lo studente, con la sua applicazione, dovrà scannerizzare i codici **QR** e il contenuto verrà inviato in una richiesta **HTTP POST** al server, il quale provvederà a **certificare** la presenza.

4.2.1. Diagramma di Collaborazione

Il diagramma seguente mostra quali sono le classi coinvolte nel caso d'uso *partecipazione lezione*. In particolare, mostra le **interazioni** necessarie per interrogare/modificare il **database** utilizzando le astrazioni fornite da Django.



4.2.2. Diagramma di Sequenza

Il diagramma seguente mostra le **interazioni** necessarie per *partecipazione lezione*. Le interazioni, dipendenti da Django, per interrogare il database, sono state omesse.