

IMPORTANTE:

questo documento potrebbe subire ancora aggiornamenti e cambiare un qualche dettaglio.

L'unica parte definitiva e' quella relativa a PYTHON/ANDROID

Progetto di laboratorio – IUM– 2021/22

Si sviluppi un'applicazione Android che gestisce **prenotazioni di ripetizioni online**

- 1 Si assuma che per ogni corso di cui si offrono ripetizioni ci siano uno più docenti alternativi tra cui scegliere con chi fare la ripetizione.
- 1 Si assuma che l'applicazione presenti le ripetizioni disponibili di una settimana pre-fissata, dal lunedì al venerdì, al pomeriggio (15.00 – 19.00). In altre parole, non è richiesto di gestire un vero calendario ma solo una griglia di slot temporali di dimensioni limitate. Ogni ripetizione disponibile è associata a un orario (dalle ore X alle ore Y del giorno Z) e ai docenti che la potrebbero tenere in quell'orario in quanto non occupati.

Seguono i dettagli richiesti per il progetto.

L'applicazione deve gestire un ruolo utente: utente registrato (cliente del servizio di ripetizioni) 2 clienti pre-registrati.

Nel dettaglio, l'applicazione deve permettere all'utente di eseguire le seguenti azioni:

- 1 Visualizzare il catalogo delle ripetizioni disponibili, specificando i docenti disponibili per ogni ripetizione prenotabile - *amministratore e cliente, interfaccia web e mobile.*
- 1 Prenotare una o più ripetizioni, tra quelle disponibili, scegliendo il corso e il docente desiderato - *cliente e amministratore*
- 1 Disdire una prenotazione specifica - Visualizzare la propria lista delle prenotazioni. La lista deve includere sia le ripetizioni ancora da fruire che lo storico delle ripetizioni prenotate in passato. Per ogni ripetizione in elenco, l'applicazione deve visualizzare lo stato della ripetizione (attiva/effettuata/disdetta) - *cliente e amministratore*
- 1

Requisiti tecnici:

- 1 Il linguaggio di implementazione può essere **Java o Kotlin o Dart**
- 1 L'applicazione deve essere basata su **architettura MVC**, con Controller + viste e Model. Si noti che non deve esserci comunicazione diretta tra viste e model: ogni tipo di comunicazione tra questi due livelli deve essere mediato dal controller.
- 1 È obbligatorio gestire le **sessioni utente**.

- 1 L'applicazione deve salvare in un **database relazionale a scelta** i seguenti tipi di informazione:
 - o account, password e ruolo degli utenti registrati;
 - o titolo dei corsi di cui si offrono le ripetizioni;
 - o nome e cognome dei docenti che tengono le ripetizioni;
 - o associazioni corso-docente;
 - o prenotazioni di ripetizioni.
- 1 L'applicazione deve controllare l'inserimento di input utente sia lato client che lato server per evitare che l'utente inserisca dati parziali o errati nei form (per esempio, per evitare che l'utente cerchi di collegarsi senza inserire login e password).
- 1 L'applicazione deve controllare che gli utenti non eseguano operazioni illecite. Per es., gli utenti non autenticati possono vedere il catalogo delle ripetizioni disponibili, ma non possono effettuare o disdire prenotazioni; deve poter vedere solo le proprie ripetizioni e non quelle altrui.

Requisiti generali dell'interfaccia utente:

- 1 L'interfaccia utente deve essere:
 - o Comprensibile (**trasparenza**). Per esempio, a fronte di errori, deve segnalare il problema; quando un'operazione viene eseguita con successo, deve visualizzare la conferma di esecuzione, a meno che la conferma non sia ridondante (in quanto il risultato si vede direttamente sull'interfaccia utente).
 - o Ragionevolmente efficiente per permettere all'utente di eseguire le operazioni con un numero minimo di click e di inserimenti di dati.
 - o In caso di errore durante l'inserimento di dati nelle form, l'interfaccia deve permettere all'utente di correggere i dati e ripetere l'operazione senza perdere i dati precedentemente inseriti (cioè, senza riempire d'accapo i moduli online).

1 Requisiti specifici per l'interfaccia utente mobile:

- o vincoli implementativi: l'applicazione deve essere scritta in Java/Kotlin usando le librerie Android. La comunicazione col lato Server avviene tramite il formato JSON.
- o **NB:** l'App deve essere sviluppata usando i componenti **native della UI di Android**. **NON** è permesso usare **Browser Web** o un loro emulatore come ad esempio "**WebView**". Nel dubbio chiedete prima.

Per gli Studenti **IUM+TWEB:**

L'interfaccia Mobile comunica col server del progetto TWEB, 1 server condiviso tra 2 client

Per gli Studenti **IUM:**

Non è necessario un Server remoto,
Il DB relazionale a vostra scelta (sqlite, etc.) può essere locale al dispositivo Android.

NB: la scelta del dominio applicativo (prenotazioni) non è vincolante e sono ammessi altri tipi di dominio purché si rispettino le specifiche funzionali e tecniche descritte sopra.

Progetto per il linguaggio python:

Si può realizzare a scelta uno dei seguenti 2 progetti:

Progetto 1:

- Immaginiamo di avere una matrice $M \times N$, di Bit. Le singole caselle possono contenere i valori 0 e 1
- questi valori formano dei pattern
- fare un programma che cerchi la presenza di specifici pattern all'interno della matrice, ad esempio delle lettere dell'alfabeto.
- i pattern possono essere ruotati di 90, 180, 270 gradi
- i pattern sono specificabili in formato a scelta dello studente.
- L'algoritmo deve essere implementato dal codice dello studente.
- Non è consentito usare librerie esterne per i pattern

Se il progetto è svolto in gruppo:

il programma avrà una GUI per inserire il pattern da cercare e visualizzare i pattern trovati

Requisito opzionale per chi si diverte:

il pattern può avere spessori maggiori di una casella

o Sono benvenute, anche se non richieste, funzionalità aggiuntive!

Progetto 2:

realizzare una interfaccia grafica in Python per il DB delle ripetizioni, funzionalmente simile a quella Android