

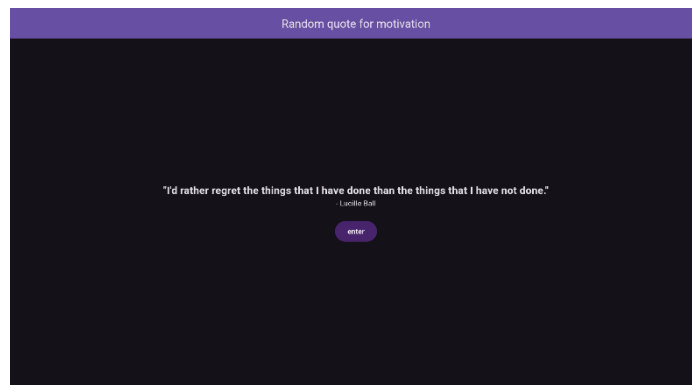
Pietro Merlo #328591

GymApp

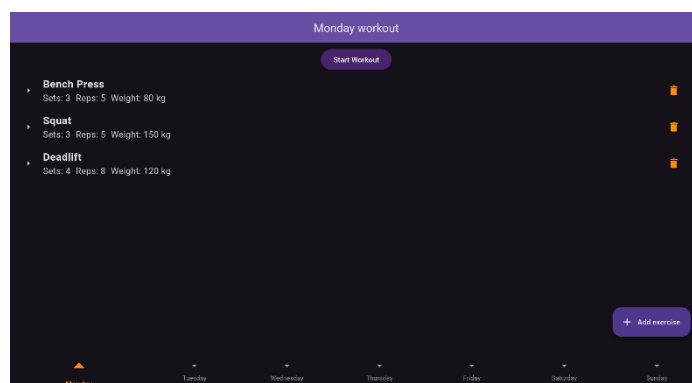
Per il progetto di Programmazione Dispositivi Mobili ed Interfacce Utente ho implementato un'applicazione Android per la palestra che permette di creare dei "Workout" per i vari giorni della settimana aggiungendo gli esercizi che si desiderano, specificandone numero di serie, di ripetizioni, peso ecc. L'applicazione poi prevede di essere utilizzata proprio durante lo svolgimento dell'allenamento proponendo all'utente l'esercizio che si deve svolgere con le relative informazioni e tenendo traccia del riposo tra una serie e l'altra con un timer.

-Flusso di Utilizzo-

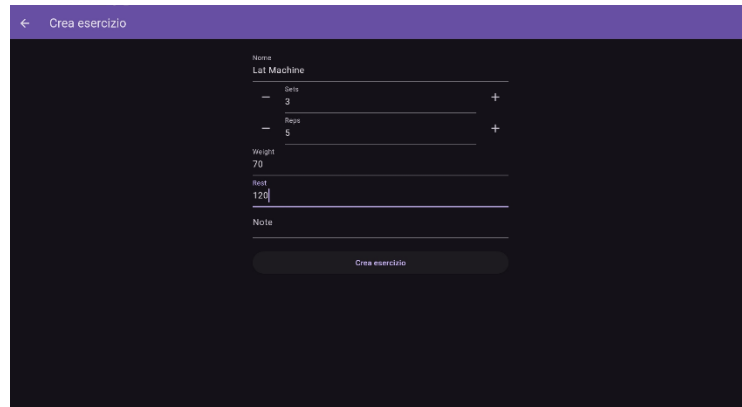
Quando l'utente clicca sull'icona e l'applicazione viene lanciata la prima schermata che verrà proposta all'utente contiene una frase motivazionale (in figura), differente ad ogni esecuzione, per intrattenere l'utente stesso mentre vengono inizializzati i *workout* creati e salvati in locale precedentemente.



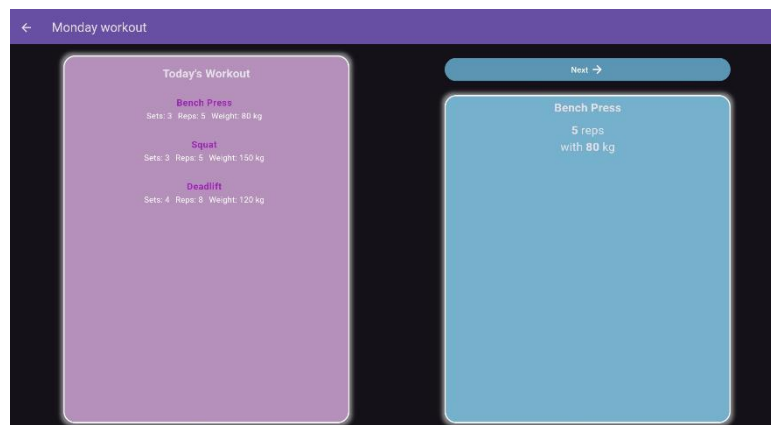
Una volta premuto il bottone "enter" si aprirà la schermata principale dell'applicazione, una schermata di tipo *BottomNavigationBar* (in figura) le cui pagine contengono i *Workout* per i vari giorni della settimana. Qualora in un'esecuzione precedente siano stati creati degli esercizi (poi non rimossi), questi ultimi resteranno presenti all'interno del *Workout* selezionato. Qualora invece il workout risulti vuoto, non sarà mostrato nessun esercizio ma un bottone che invita l'utente a crearne uno.



Nel momento in cui l'utente, tramite il pulsante *"create exercise"* del workout vuoto o tramite il pulsante *"Add exercise"* presente per tutte le pagine della *BottomNavigationBar* decide di aggiungere un esercizio al *Workout* selezionato si aprirà una pagina contenente un form contenente tutti i campi necessari (in figura). Una volta completata la compilazione dei vari campi (alcuni obbligatori come ad esempio nome, serie e ripetizioni, altri facoltativi come le eventuali note), diventa possibile premere il pulsante *"create exercise"* il quale aggiunge l'esercizio appena creato al workout selezionato e rimanda alla pagina con la *BottomNavigationBar*.



Una volta aggiunti tutti gli esercizi ad un determinato *workout* è possibile iniziare l'allenamento tramite il pulsante *"start workout"* il quale una volta premuto aprirà una pagina (in figura) che prevede un riassunto del *workout* appena iniziato e una finestra che propone all'utente l'esercizio da svolgere con le relative informazioni e a seguire il riposo prima della serie successiva tramite un timer circolare.



-Struttura dati-

I dati relativi ai vari esercizi sono contenuti nell'oggetto di tipo *Exercise*, una lista di *Exercise* è poi contenuta nell'oggetto di tipo *Workout*. Per i vari giorni della settimana è stato creato un *NotifierProvider<Workout, List<Exercise>>* specifico. Per la pagina che mostra l'allenamento è stata creata una struttura dati ad hoc chiamata *Event*, la quale può contenere sia eventi di tipo esercizio che di tipo riposo. Una *List<Event>* è stata utilizzata tramite uno *StateProvider* per gestire il susseguirsi di esercizi e riposi durante lo svolgimento dell'allenamento.

-Salvataggio in locale-

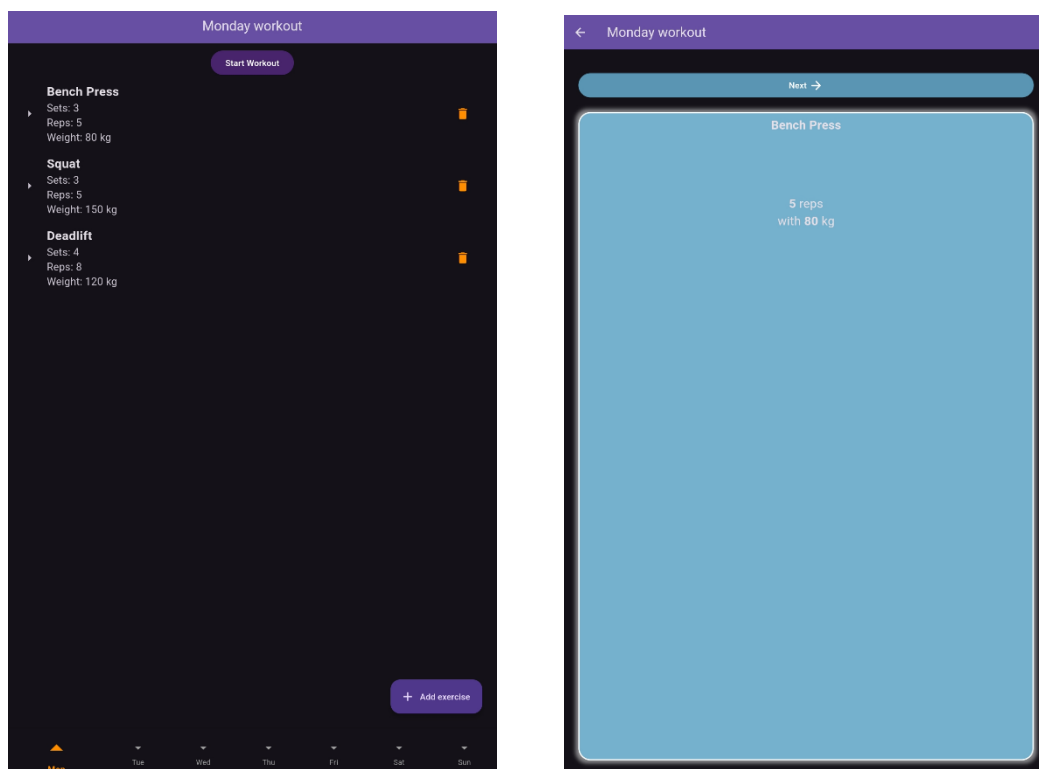
Per mantenere localmente i dati, ogni volta che un esercizio viene aggiunto ad un workout, l'intero workout viene salvato all'interno della cartella locale dell'applicazione in formato Json all'interno di un file con il nome appropriato. Come accennato in precedenza, all'avvio dell'applicazione i dati salvati localmente, qualora presenti, vengono letti dall'applicazione e inseriti all'interno dei rispettivi Workout.

-Comunicazione in remoto-

La comunicazione in remoto è stata utilizzata per fornire, all'avvio, mentre i dati degli allenamenti vengono inizializzati dallo storage locale, una frase motivazionale ottenuta tramite l'indirizzo: <https://api.quotable.io/quotes/random> il quale restituisce sempre in formato Json una frase differente ogni volta assieme ad ulteriori informazioni come id, Autore, tags, date di aggiunta e modifica. Al fine del display della frase sono stati gestiti solamente il nome dell'autore e il corpo della frase stessa.

-Interfaccia grafica responsiva-

L'applicazione è stata sviluppata per dispositivi Android. La responsività dell'interfaccia utente è stata dunque implementata per la rotazione dello schermo con due modalità: *Landscape* e *Portrait*. La pagina che contiene la *BottomNavigationBar*, ad esempio, mostra i giorni della settimana in basso sotto forma di nome intero o di sigla a seconda dell'orientamento dello schermo (per una questione di spazi, in modalità *Portrait* sono state utilizzate sigle come "Mon", "Tue" e così via). Similmente, le informazioni dei vari esercizi sono mostrate in riga nel caso dell'orientamento orizzontale e in colonna in caso contrario. Per quanto riguarda la pagina dell'allenamento, in modalità *Landscape* prevede il riassunto sulla sinistra e il display del susseguirsi degli eventi sulla destra, in modalità *Portrait* prevede solamente il display degli eventi. (figure sottostanti)



-Pacchetti aggiuntivi-

Per lo sviluppo di questa applicazione sono stati utilizzati alcuni pacchetti aggiuntivi come i provider Riverpod (*flutter_riverpod*: ^2.5.1) per mantenere lo stato interno dell'applicazione senza dover utilizzare widget di tipo *Stateful*.

Per la pagina di creazione dell'esercizio è stato utilizzato il pacchetto *flutter_form_builder*: ^9.3.0 che facilita la creazione e l'utilizzo di form come quello presente all'interno della pagina.

Il pacchetto *path_provider*: ^2.1.3 è stato utilizzato per ottenere il path della cartella in locale dove salvare i dati.

Per un'implementazione semplice e funzionale del timer nella pagina d'allenamento è stato utilizzato il pacchetto *circular_countdown_timer*: ^0.2.3.

Il pacchetto *uuid*: ^4.4.0 è stato utilizzato per fornire id univoci agli esercizi aggiunti ai vari workout.

Infine, il pacchetto *http*: ^1.2.1 è stato utilizzato per l'implementazione della comunicazione remota nella pagina di startup.