



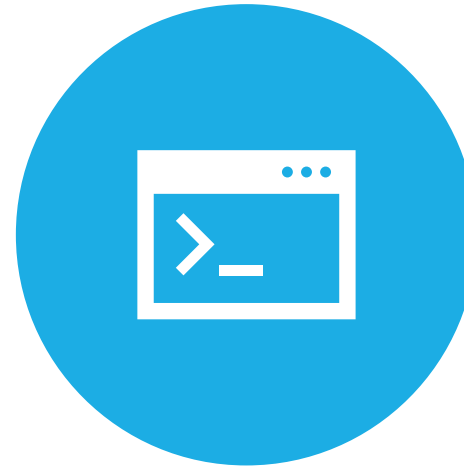
TECNOLOGIE MOBILI E CLOUD

Homework #4 - Gruppo LIM – Luca Lorenzi, Longo Dandis Iana, Miriam Kenna

PRESENTAZIONE HOMEWORK #4



OBIETTIVI



IMPLEMENTAZIONE

OBIETTIVI

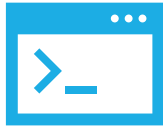
OBIETTIVI PER IL HOMEWORK #4

1. Aggiungere funzionalità visualizzazione classifica di categoria
2. Migliorare UI/UX
3. Aggiungere pulsante/gesture per ricaricare i dati

BONUS TASK

4. Aggiungere gestione della griglia di partenza
5. Aggiungere visualizzazione dei risultati di tutti i concorrenti di un club

IMPLEMENTAZIONE



OBIETTIVO #1

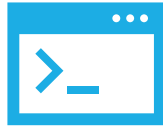
DESCRIZIONE:

- Aggiungere funzionalità visualizzazione classifica di categoria

STEP:

- Collegamento al API Gateway:
`https://s96his1myh.execute-api.us-east-1.amazonaws.com/list_races/list_races?id=1'`
- Con il `jsonDecode` elaboriamo i dati che vengono passati nel body della risposta, in attesa del caricamento aggiungiamo il widget `CircularProgressIndicator()` che fa apparire un cerchio di caricamento intanto che future che contiene i dati arriva a compimento
- I dati vengono salvati in una lista, con un ciclo `for` esploriamo gli elementi della lista e salviamo in un'altra lista i nomi delle gare, poi con il metodo `«toSet().toList();»` eliminiamo i duplicati
- Infine proiettiamo sullo schermo con un altro ciclo `for` gli elementi salvati nella nuova lista, otteniamo così la lista delle gare

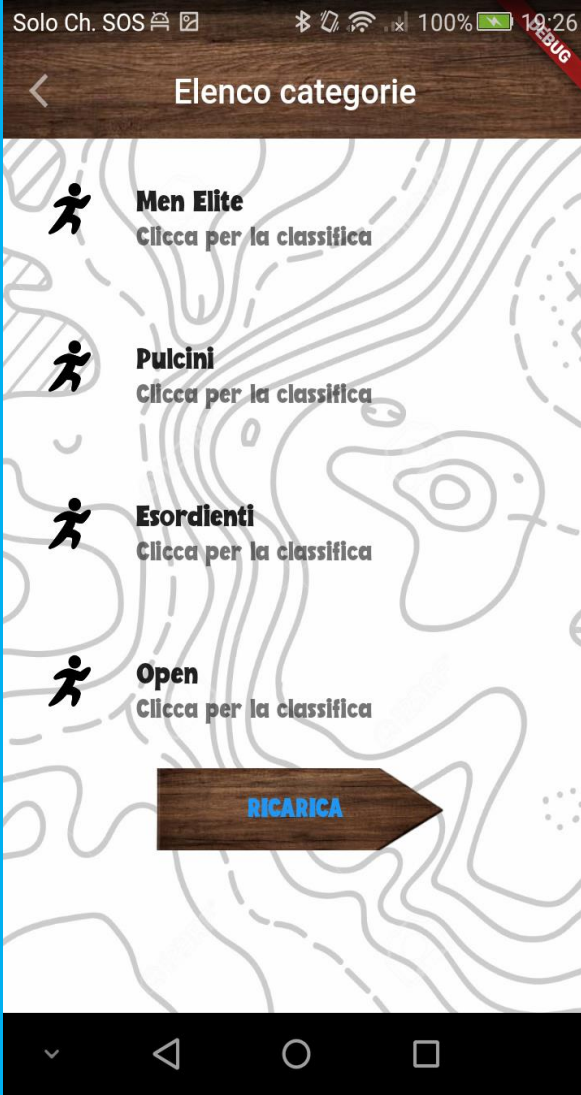
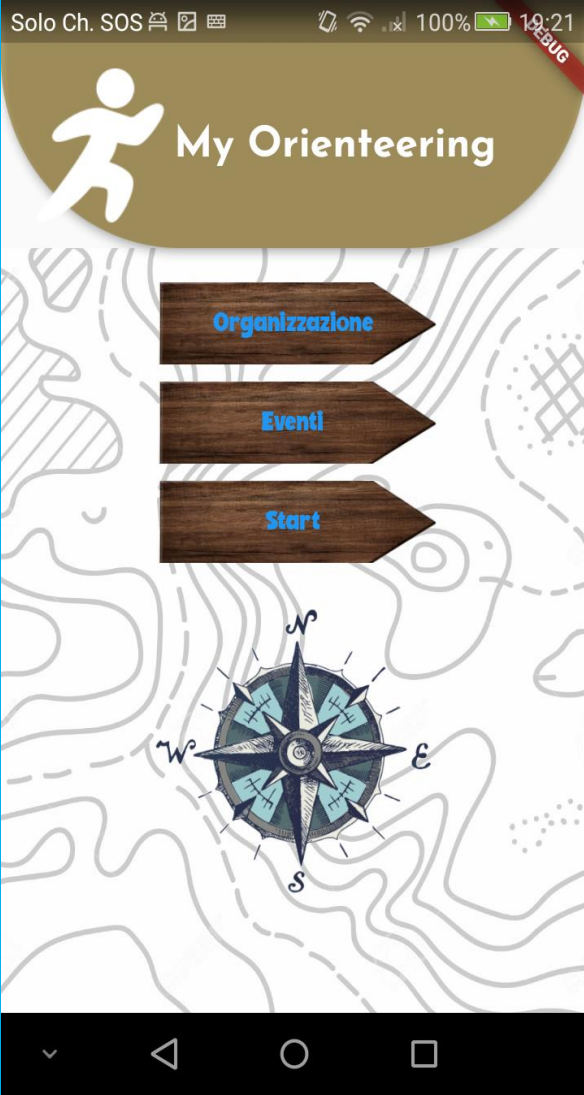
IMPLEMENTAZIONE



OBIETTIVO #1

STEP:

- Cliccando sul nome della gara, passiamo il parametro «nome_gara» alla pagina nuova che conterrà le categorie, con il parametro andiamo ad esplorare di nuovo la lista e selezioniamo solo le categorie appartenenti a quella gara. L'esplorazione avviene con un ciclo for. Tramite esso andiamo a creare due liste:
 1. La prima contiene soltanto le stringhe delle categorie, eliminati i duplicati
 2. La seconda contiene le stringhe complete dei partecipanti all'evento, passato come parametro precedentemente.
- Per visualizzare le categorie utilizziamo un listTile passandoli come parametro la prima lista.
- A questo punto cliccando sulla categoria viene passata come parametro la seconda lista alla nuova pagina, insieme alla categoria
- Infine per creare la classifica, andiamo a creare un'altra lista che contiene i partecipanti solo di quella categoria, visualizzati in ordine di posizione grazie al metodo «sort»



IMPLEMENTAZIONE

OBIETTIVO #2

DESCRIZIONE:

- Migliorare UI/UX

STEP:

- **Per migliorare l'aspetto grafico dell'APP abbiamo:**

1. Usato nella Home Page una gif animata di una bussola;
2. Sempre nella home abbiamo personalizzato l'AppBar;
2. Implementato dei font personalizzati: BRONSON.ttf;
3. I bottoni con il background di un'immagine che ricordano i tipici cartelli che si possono incontrare in montagna;
4. Aggiunto icone personalizzate;
5. In tutte le pagine abbiamo personalizzato il background della schermata, utilizzando una mappa
6. Inoltre ogni pagina riporta nell'AppBar il tasto per ritornare alla pagina precedente

IMPLEMENTAZIONE

OBIETTIVO #3

DESCRIZIONE:

- Aggiungere pulsante/gesture per ricaricare i dati

STEP:

- Il pulsante per la ricarica è un textbutton che una volta cliccato fa una nuova chiamata get al API Gateway del AWS e riceve una lista aggiornata che poi potrà essere filtrata con i parametri di interesse, visualizzando i dati aggiornati

SOLUZIONI



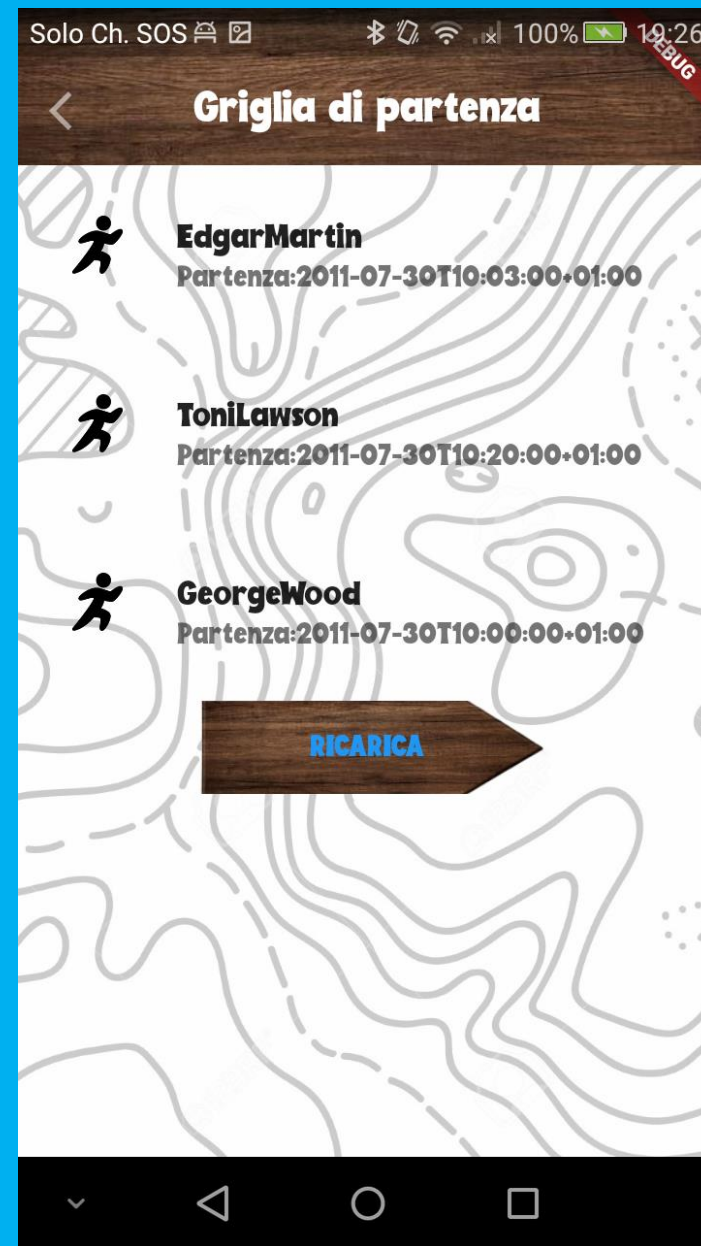
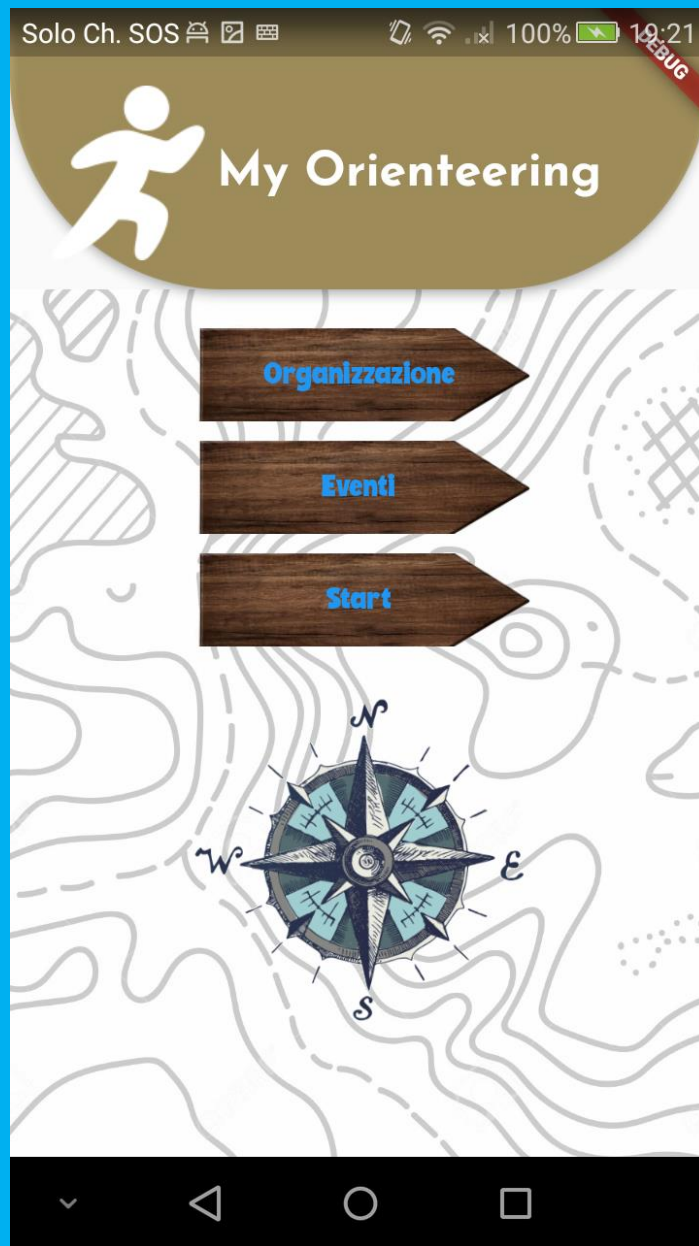
OBIETTIVO #4

DESCRIZIONE:

- Aggiungere gestione della griglia di partenza

STEP:

- Facciamo una chiamata get al API Gateway che collegandosi alla funzione lambda ci ritorna come risultato il json dei dati della «StartList1.xml» precedentemente inserita con un'altra funzione lambda all'interno del dynamoDB
- Successivamente viene creata un nuovo listTile che proietta nome, cognome del partecipante e la data della partenza con il corrispettivo orario , per tutti i partecipanti della lista



SOLUZIONI



OBIETTIVO #5

DESCRIZIONE:

- Aggiungere visualizzazione dei risultati di tutti i concorrenti di un club

STEP:

- Collegamento al API Gateway:
`https://s96his1myh.execute-api.us-east-1.amazonaws.com/list_races/list_races?id=1`
- Con il `jsonDecoden` elaboriamo i dati che vengono passati nel body della risposta, in attesa del caricamento aggiungiamo il widget `CircularProgressIndicator()` che fa apparire un cerchio di caricamento intanto che future che contiene i dati arriva a compimento
- I dati vengono salvati in una lista, con un ciclo `for` esploriamo gli elementi della lista e salviamo in un'altra lista i nomi dei vari club, poi con il metodo `«toSet().toList();»` eliminiamo i duplicati
- Infine proiettiamo sullo schermo con un altro ciclo `for` gli elementi salvati nella nuova lista, otteniamo così la lista con l'elenco delle organizzazioni

SOLUZIONI



OBIETTIVO #5

STEP:

- Cliccando sul nome del club, passiamo il parametro «person_organisation_name» alla nuova pagina che conterrà i concorrenti, con il parametro andiamo ad esplorare di nuovo la lista e selezioniamo solo i partecipanti appartenenti a quel club. L'esplorazione avviene con un ciclo for. Tramite esso andiamo a creare la lista che contiene il nome e il cognome dei concorrenti
- A questo punto cliccando sul concorrente viene passato come parametro la lista ed un indice corrispondente alla posizione del partecipante nella lista
- Infine viene proiettato sullo schermo la categoria e la posizione del concorrente del indice corrispondente

