Contents

List of Tables

0.1 Introduzione

Ad oggi la maggior parte delle persone che si organizzano per andare insieme in montagna fanno ciò attraverso passaparola, gruppi Telegram o gruppi Whatsapp. Manca quindi un modo per aggregare queste persone e permettergli di comunicare in maniera agevole e veloce per organizzare le escursioni che vogliono fare. L'obiettivo è quello di creare una app che permetta di fare proprio questo, mettendo a disposizione degli utenti un ambiente dove poter pubblicare escursioni che intendono compiere e permettendo ad altri utenti di iscriversi per prendervi parte. Gli attori coinvolti saranno essenzialmente di 2 tipi: Organizzatori: possono creare degli eventi e pubblicarli sulla piattaforma; Partecipanti: possono unicamente iscriversi agli eventi pubblicati dagli organizzatori e non possono crearne di loro. Sia organizzatori che partecipanti possono iscriversi a degli eventi pubblicati sulla piattaforma. Per tutti gli attori andranno salvate le informazioni anagrafiche di base (eg. nome e cognome) e un'immagine profilo. Sarà poi necessario poter distinguere a quale categoria di attore appartiene un dato utente, in modo da poter personalizzare l'esperienza sul client fornendo agli organizzatori i comandi per creare un evento. Le escursioni conterranno svariate informazioni sull'evento come l'ora e il giorno, oltre alla posizione. Dovranno inoltre contenere una lista di utenti iscritti e una seconda di organizzatori, dato che un evento può avere più organizzatori. Sarà inoltre necessario specificare un'ora e un luogo di ritrovo, la difficoltà del percorso e l'attrezzatura necessaria (per esempio se sono presenti pareti da scalare) o consigliata (per esempio una quantità minima di acqua). Lato client è inoltre richiesto che ogni evento mostri le condizioni meteorologiche attese per il giorno dell'evento, tale informazione verrà recuperata tramite le API di Visual Crossing. La piattaforma è creata per rasentare un social media, dove gli organizzatori hanno dei loro profili che possono essere seguiti dagli utenti. Inoltre, al termine di ogni escursione, viene data all'utente la possibilità di esprimere un giudizio da 1 a 5 stelle all'escursione. La media di questi punteggi determinerà un giudizio in stelle assegnato all'organizzatore. Ogni attore disporrà inoltre di una lista di escursioni a cui ha preso parte, le informazioni contenute in questa lista potranno essere usate per consigliargli altre escursioni in base

a criteri come il giorno, la difficoltà e il dislivello. Ogni utente dovrà registrarsi alla piattaforma tramite il client e dovrà fare l'accesso prima di poter visualizzare gli eventi. Nel momento dell'iscrizione verrà fornito un campo per permettere agli organizzatori di inserire una speciale chiave fornita dall'organizzazione responsabile della piattaforma. Questa gli permetterà di iscriversi come organizzatori e non come semplici partecipanti. Prevediamo inoltre un limite massimo di persone per ogni escursione, che verrá impostato dall'organizzatore. Nel momento in cui il numero di iscrizioni supera il numero massimo di persone consentite un algoritmo automatico scarta coloro che ritiene non essere all'altezza del percorso, basandosi sulle loro precedenti esperienze. Un esempio potrebbe essere un utente che non ha mai avuto esperienze su un percorso ad alta difficoltà.

0.2 Requisiti e casi d'uso

0.2.1 Use cases

Di seguito sono riportati i requisiti del sistema che si vuole implementare. Per ciascun caso si indica il codice del caso d'uso, il titolo e la user story al fine di facilitare la comprensione.

- **UC1 - Login**. Come utente voglio accedere al mio account
- **UC2 - Signup**. Come utente voglio potermi registrare al sistema.
- **UC3 - Logout**. Come utente voglio potermi disconnettere dal sistema.
- **UC4 - Visualizzazione iscrizioni**. Come utente voglio visualizzare gli eventi ai quali sono iscritto.
- **UC5 - Visualizzazione meteo**. Come utente voglio visualizzare il meteo entro 5 giorni dall'evento.
- **UC6 - Visualizzazione consigliati**. Come utente voglio poter visualizzare gli eventi consigliati.
- **UC7 - Ricerca eventi**. Come utente voglio poter cercare dei nuovi eventi, scegliendo degli opportuni parametri di ricerca.
- **UC8 - Visualizzazione risultati ricerca.** - Come utente, voglio poter visualizzare tutti i risultati della ricerca.
- **UC9 - Visualizzazione specifico risultato.** - Come utente, voglio poter consultare il singolo risultato della ricerca eseguita, visualizzando nei dettagli informazioni quali la località, la data e l'organizzatore.
- **UC10 - Iscrizione ad un nuovo evento.** Come utente, voglio potermi iscrivere ad uno degli eventi disponibili.
- **UC11 - Cancellazione iscrizione.** Come utente, voglio poter cancellare la mia iscrizione ad un evento.

- **UC12 - Visualizzazione partecipanti.** Come utente, voglio poter visualizzare gli utenti che parteciperanno ad un evento.
- **UC13 - Visualizzazione profilo utente.** Come utente, voglio poter visualizzare sul mio profilo le esperienze a cui ho preso parte e quelle a cui sono iscritto.
- **UC14 - Visualizzazione profilo organizzatore.** Come utente organizzatore, voglio poter visualizzare sul mio profilo gli eventi passati da me organizzati e quelli futuri.
- **UC15 - Algoritmo iscrizione.** Come utente, voglio che solo i profili più adeguati possano partecipare ad un evento, e che solamente gli N profili più adeguati vengano accettati.
- **UC16 - Dettagli evento.** Come utente, voglio poter visualizzare tutte le informazioni relative all'evento per poter valutare se iscrivermi o meno.
- **UC17 - Eliminazione evento.** Come organizzatore voglio poter cancellare un evento che ho creato.
- **UC18 - Creazione evento.** Come organizzatore voglio poter creare un evento.

0.2.2 Priorità dei casi d'uso

I casi d'uso possono essere ripartiti all'interno di tre code, a seconda della loro priorità nel processo di sviluppo.

- Coda ad alta priorità: contiene i requisiti ritenuti fondamentali per il funzionamento dell'applicativo, ossia la creazione dei profili utente, la creazione di nuovi eventi ed il sistema di iscrizione agli eventi stessi.

- Coda a media priorità: coda contenente funzionalità di supporto, principalmente legate alla visualizzazione di informazioni ulteriori riguardo agli utenti e/o agli eventi.

- Coda a bassa priorità: coda nella quale vengono indicate le funzionalità meno rilevanti, o che verranno implementate in iterazioni future.

In particolare, nelle tabelle di seguito [1,2,3] è possibile osservare come i casi d'uso precedentemente individuati possano essere suddivisi.

NUMERO	TITOLO
UC1	Login
UC2	Signup
UC3	Logout
UC18	Creazione evento
UC17	Eliminazione evento
UC10	Iscrizione ad un nuovo evento
UC11	Cancellazione iscrizione
UC7	Ricerca eventi
UC8	Visualizzazione risultati ricerca
UC9	Visualizzazione specifico risultato

Table 1: Coda ad alta priorità

NUMERO	TITOLO
UC5	Visualizzazione meteo
UC6	Visualizzazione consigliati
UC15	Algoritmo iscrizione
UC13	Visualizzazione profilo utente
UC14	Visualizzazione profilo organizzatore

Table 2: Coda a media priorità

NUMERO	TITOLO
UC4	Visualizzazione iscrizioni
UC12	Visualizzazione partecipanti
UC16	Dettagli evento

Table 3: Coda a bassa priorità

0.2.3 Use case diagram

I requisiti precedentemente elencati possono essere visualizzati graficamente tramite l'use case diagram riportato in figura.

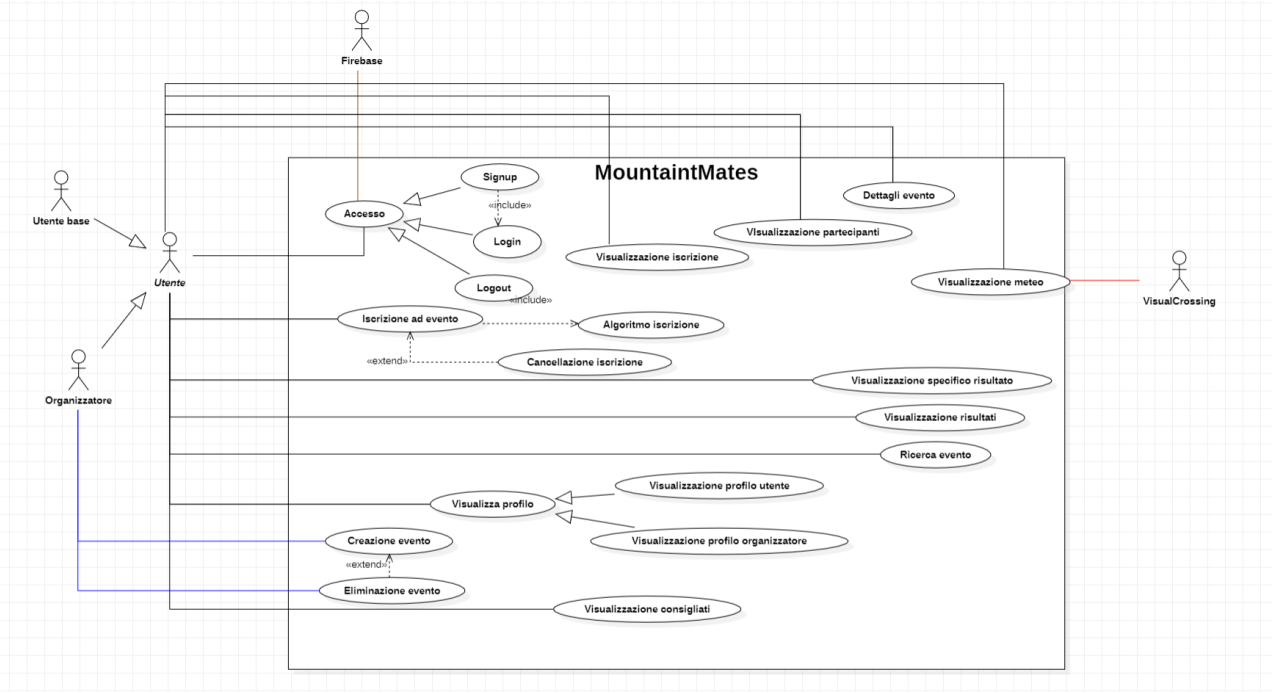

Figure 1: Use cases

E' stata introdotta una generalizzazione sugli utenti per distinguere le funzionalità accessibili da un utente base rispetto a quelle di un organizzatore. I due attori Firebase e VisualCrossing sono coinvolti nei casi d'uso relativi all'autenticazione ed alla visualizzazione delle condizioni meteo sul luogo dell'evento.

Inoltre, l'iscrizione ad un evento richiede l'esecuzione del caso d'uso algoritmo di iscrizione in quanto, come detto precedentemente, una volta raggiunto un massimo di iscritti solamente i primi N più qualificati saranno accettati all'evento.

0.3 Architettura

L'architettura del sistema è stata definita attraverso il deployment diagram mostrato in figura 2, espresso in notazione free style. Si identifica un pattern architetturale three-tier, dove:

1. Il Presentation Layer è costituito dalle applicazioni lato client che vengono utilizzate dagli utenti e hanno il solo compito di recuperare le informazioni dal sistema e visualizzarle;

2. L'Application Layer gestisce la logica di business. È costituito da due macro-sistemi specializzati:

   - **Gestore profilo**, responsabile di tutto ciò che riguarda i profili degli utenti, come gestire la registrazione, il login e soddisfare le query inerenti all'utente. Questo gestore si connette ai servizi di Firebase per gestire l'autenticazione in fase di login;
   - **Gestore escursione**, responsabile di gestire le richieste inerenti alle escursioni, come ottenere una lista o i dettagli di uno specifico evento. Per fornire informazioni meteorologiche riguardo agli eventi, il gestore si connette ai servizi di Visual Crossing tramite API call. La risposta viene elaborata in loco e restituita al client in un formato più breve da definirsi in futuro;

3. Il Data Layer si occupa della persistenza dei dati ed è costituito da un database dove sono conservate le informazioni di ogni evento e utente registrato alla piattaforma.

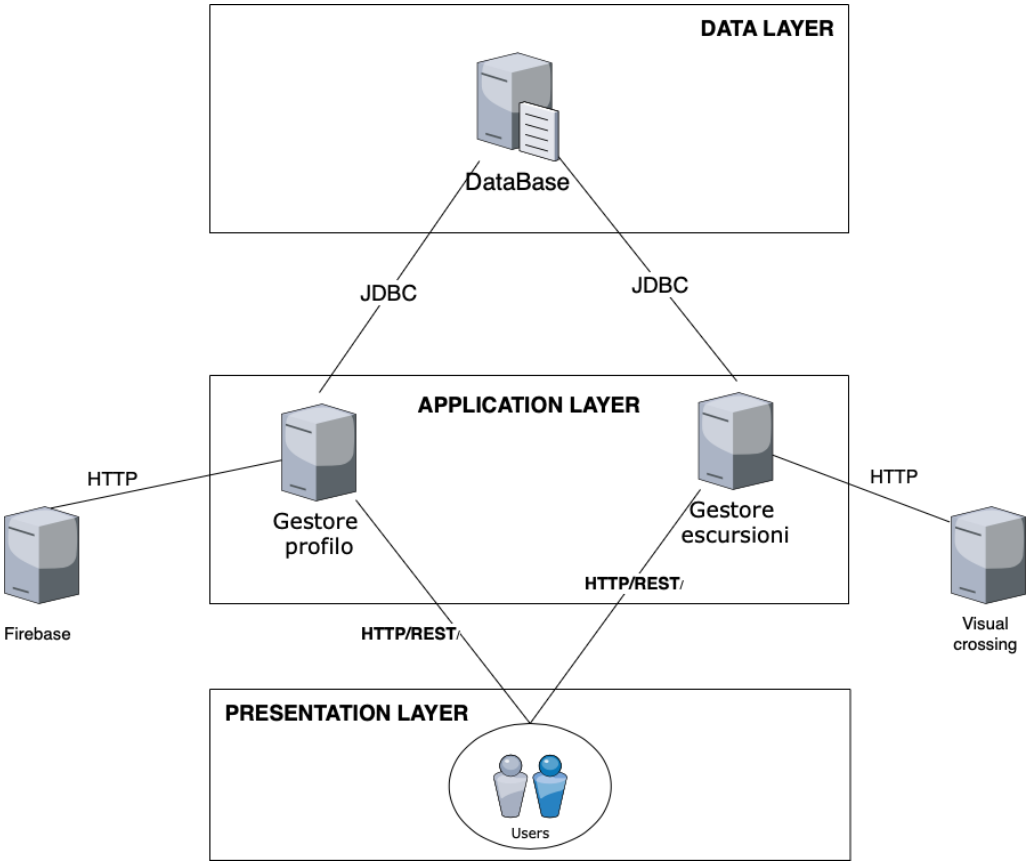

Figure 2: Deployment Diagram in versione free style

0.4 Toolchain

Per la realizzazione della piattaforma verranno utilizzati i seguenti strumenti:

- **Modellazione**:
  - Diagramma dei casi d'uso, deployment diagram, component diagram, class diagram, diagrammi di flusso, diagramma entità-relazione: Draw.io.
- **Implementazione Applicazione Client**:
  - Linguaggio di programmazione: Dart;
  - IDE: Visual Studio Code;
  - Interfaccia grafica: Flutter;
  - Analisi statica: Dart analyze;
  - Analisi dinamica: Dart analyze.
- **Implementazione Web Server**:
  - Linguaggio di programmazione: Java;
  - IDE: Eclipse;
  - Framework: Spring;
  - Analisi statica: STAN4J;
  - Analisi dinamica: JUnit;
  - Deployment: Render.
- **Implementazione Database**:
  - Tipologia: relazionale;
  - Database: Posgress;
  - Provider: Render.
- **Documentazione, versioning e organizzazione del team**:
  - Documentazione: Latex;

- Versioning: GitHub;
- Git client: Github Desktop;
- Organizzazione del Team: Microsoft Teams e Trello.