Contents

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 Casi d'uso

0.3 Architettura

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.