UNIVERSITÀ MILANO BICOCCA

Corso di Ingegneria del software
2022/23

Relazione di Progetto "Travel Time Aware"

Davide Creati

Introduzione:

Travlendar è una applicazione che consiste in un calendario personalizzabile con funzioni automatizzate.
La funzione principale permette di tenere in considerazione gl spostamenti tra gli eventi inseriti dall'utente, supportando diversi tipi di mezzi di trasporto.
Ogni utente può selezionare i propri mezzi di trasporto (ad esempio in base alle disponibilità). Inoltre può selezionare dei vincoli sui mezzi in base ai tempi di percorrenza.

Implementazione:

a livello implementativo è stato usato il framework spring boot per il lato backend mentre il frontend è stato usata la libreria FullCalendar per la grafica e motore del calendario insieme a jquery

Sono stati tenuti in considerazione diversi design pattern e valutato il codice tramite strumenti come sonarqube ed understand.

Il primo pattern preso in considerazione è il Factory. La classe Interval.java contiene la specifica per rappresentare un generico intervallo tra due orari (espressi nel formato dd:MM:yyyy hh:mm:ss).
Il calendario in sè salva all'interno degli oggetti Schedule che sono composti da un oggetto Event, che rappresenta l'evento inserito dall'utente, e un oggetto Travel che rappresenta il viaggio tra tra l'evento correlato e quello prima.
Travel viene implementato tramite il pattern state in quanto un evento può avere o meno un altro evento prima. In caso l'evento inserito sia il primo della giornata, il Travel sarà di tipo EmptyTravel in quanto non ci sarà uno spostamento da un evento ad un altro. Altrimenti se il ci sono eventi prima di quello inserito il Travel si troverà nello stato di ExistTravel che comporterà il mezzo da usare per lo spostamento.

Per la gestione dell'applicazione nella sua completezza viene implementato il patter architetturale MVC per separare la logica di business da quella di visualizzazione. Viene implementato il pattern controller per la gestione delle richieste ai veri endpoint.

Inoltre viene fatto uso del pattern pure fabrication tramite l'utilizzo di una classe service per gli utenti, utilizzata in nei processi di autenticazione.

Analisi con sonarqube e understand: L'obiettivo principale con sonarqube è stato quello di raggiungere un grado di valutazione di almeno "A". Uno dei problemi riscontrati era un Security Hotspots in quanto era presente un metodo che utilizzava "ThreadLocalRandom" soggetto a problemi di sicurezza cui la possibilità di calcolare il seed di generazione del valore associato. Un altro errore trovato era la regular expression contenuta in User per la validazione dell'email che, per com'era precedentemente, poteva generare dei problemi di stack overflow per input grandi.

Per quanto riguarda le analisi con understand uno degli antipattern trovati è la God Class (riferita al CalendarController) in quanto contiene molti metodi non-cohesive. Per mancanza di tempo non è stato possibile aggiustare questo problema.