

Progetto Ingegneria Software

NomadBees

Gruppo T-37:
Matteo Pontalti, Matteo Bregola,
Riccardo Libanora.

Architettura

Contenuti

Contenuti.....	2
Scopo del Documento	3
Diagramma delle Classi	3
Diagramma Complessivo.....	3
Utente	4
Login e Registrazione	4
Viaggio, Tappe e Visualizza Viaggio.....	4
Mappa	5
Ricerca Viaggi e Profili	5

Scopo del Documento

Il seguente documento presenta l'architettura del progetto NomadBees tramite l'utilizzo del diagramma delle classi in UML e codice OCL. La realizzazione di questo materiale è frutto del lavoro eseguito nei precedenti documenti, risulta quindi necessario avere una conoscenza di quest'ultimi per poter comprendere in maniera più efficace le decisioni che sono state prese.

Diagramma delle Classi

In questa sezione sarà presentato e spiegato il diagramma delle classi del progetto. La prima immagine rappresenta il diagramma completo che permette un immediata visione d'insieme e poi successivamente verranno precisate le classi con i propri attributi e metodi. Gli attributi, locati nella porzione superiore della classe identificano i dati che la classe gestisce i quali possono provenire da interazioni con gli utenti, dal database o dati di "supporto" generati dalla classe stessa. I metodi invece sono le funzionalità che la classe utilizza internamente per gestire i suoi dati o che fornisce al sistema. Ogni classe amministra un particolare aspetto del progetto, l'unione di esse fornisce tutte le funzionalità esposte nei documenti precedenti. Affinché ciò avvenga è necessario che vi sia un dialogo tra le classi nella gestione dei dati: da qui nascono le associazioni presenti nel diagramma. Le associazioni garantiscono la comprensione delle interrelazioni tra le classi.

Il diagramma è stato creato tramite l'Unified Modeling Language (UML).

Diagramma Complessivo

Utente

La classe Utente e Viaggio sono le due principali fondamenta del progetto *NomadBees*. Analizzando il diagramma delle componenti abbiamo notato che ogni componente gestiva dati riguardanti gli utenti o i viaggi. Il diagramma quindi è stato realizzando partendo dall'idea di mettere al centro queste due classi.

La classe utente contiene come attributi i dati collezionati durante la registrazione e l'utilizzo del sito da parte di un suo utilizzatore. Vi è inoltre un attributo "autenticato" che è stato pensato per garantire specifiche funzionalità per queste tipologie di utenti e può essere modificato dalle classi Login e Registrazione.

Per quanto concerne i metodi, sono racchiuse nella classe tutte le funzionalità che permettono la modifica o l'eliminazione degli attributi sopra riportati. E' fatta eccezione per quanto riguarda l'attributo "viaggi" poiché nonostante esso sia parte delle informazioni che si hanno sull'utente, per una questione di complessità, viene gestito dalla classe "Viaggio."

Login e Registrazione

La Classe "Login" si occupa di permettere ad un utente di autenticarsi, richiedendo a quest'ultimo l'username del profilo e la password.

La classe "Registrazione" gestisce la registrazione al sito, necessità di inserire specifici dati dei quali valuta la correttezza tramite tre metodi.

Viaggio, Tappe e Visualizza Viaggio

Viaggio contiene i dati che verranno salvati quando si crea un post. Il viaggio è costituito, come indicato dall'associazione, da tappe. Durante la creazione di un post si possono aggiungere due o più tappe che contengono foto, descrizione e posizione.

E' stata pensata una classe dedicata per la visualizzazione del viaggio che gestisce quando generare la mappa o cosa inserire nella preview.

Mappa

La classe Mappa si occupa di gestire la creazione della mappa per quando si visualizza un viaggio. I suoi attributi e metodi fanno riferimento a quelli utilizzati dalla Api di Google Maps.

Ricerca Viaggi e Profili

Le seguenti due classi si occupano di gestire le funzionalità del sito di ricerca. Potevano essere inserite come parte della classe viaggio ed utente ma è stato scelto di creare due classi separate per due motivi. Come prima cosa, essendo complesse, renderle indipendenti riduce la complessità delle classi anche in vista di possibili future nuove funzionalità del sito. Inoltre inserirle come parte di altre classi avrebbe ridotto la coesione interna delle classi, elemento base della loro organizzazione.