Progetto e sviluppo di un'applicazione mobile per il calcolo dei percorsi

Candidato: Daniele Cristofori
Relatore: Prof. Sonia Bergamaschi
Correlatore: Prof. Riccardo Martoglia

Motivazioni

Obiettivi

- Migliorare la vita dei cittadini e la loro mobilità usando Google Maps ma fornendo delle funzionalità in più

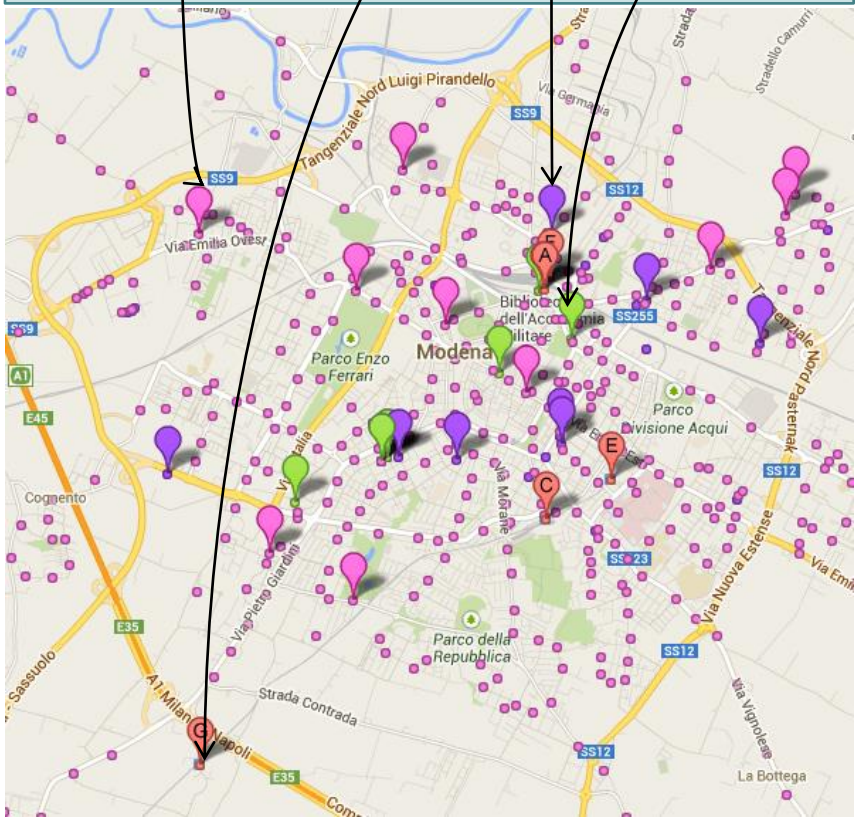
1. Consentire l'utilizzo di una combinazione di mezzi di trasporto quali la bicicletta, i mezzi pubblici e l'automobile (autonoleggio) promuovendo di fatto la Green Economy
2. Dare la possibilità di impostare un tempo massimo di arrivo a destinazione

- Per questo è stata realizzata un'applicazione per iPhone ("Route Planner")

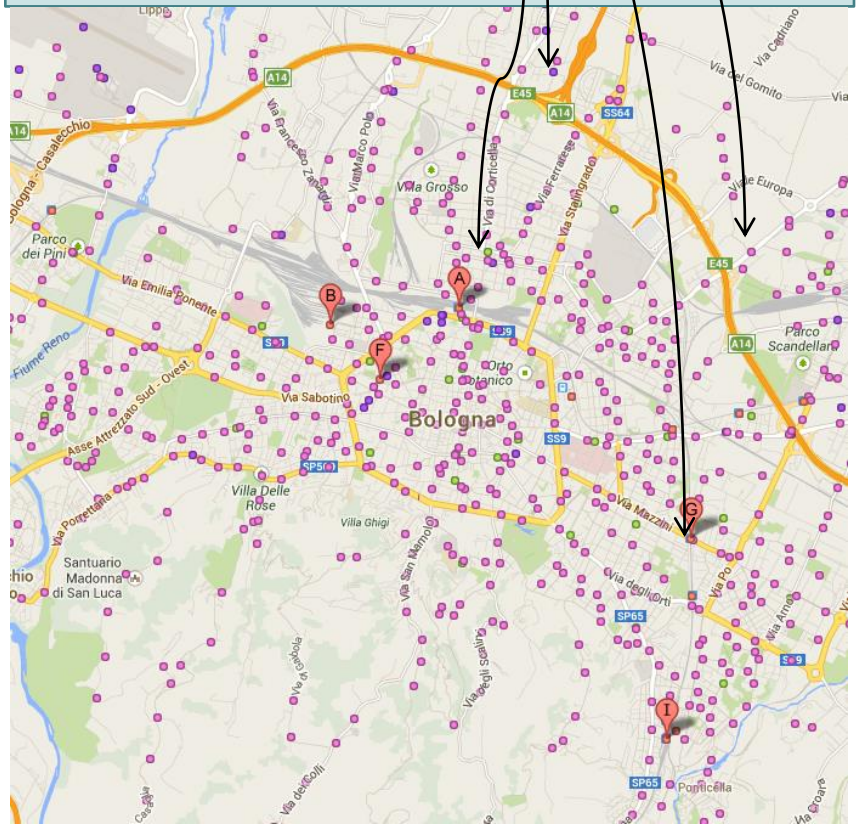
Dati Google Maps

- Bus
- Treno
- Autonoleggio
- Postazione taxi

Modena

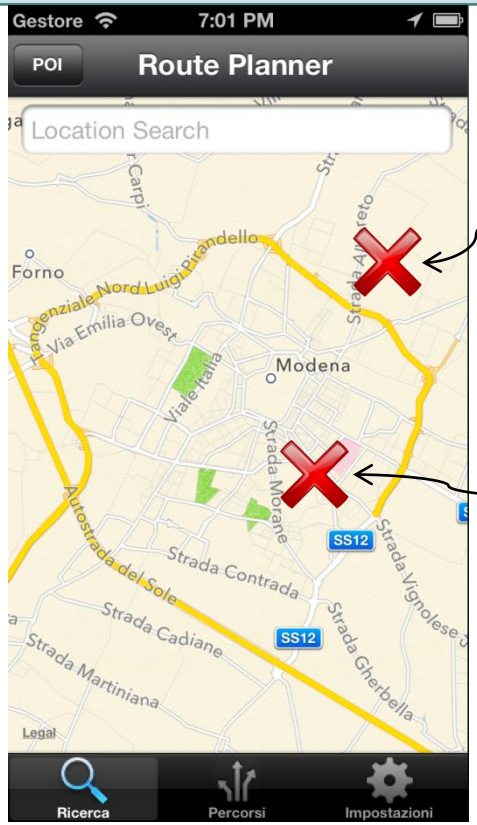


Bologna



Open Data Emilia-Romagna

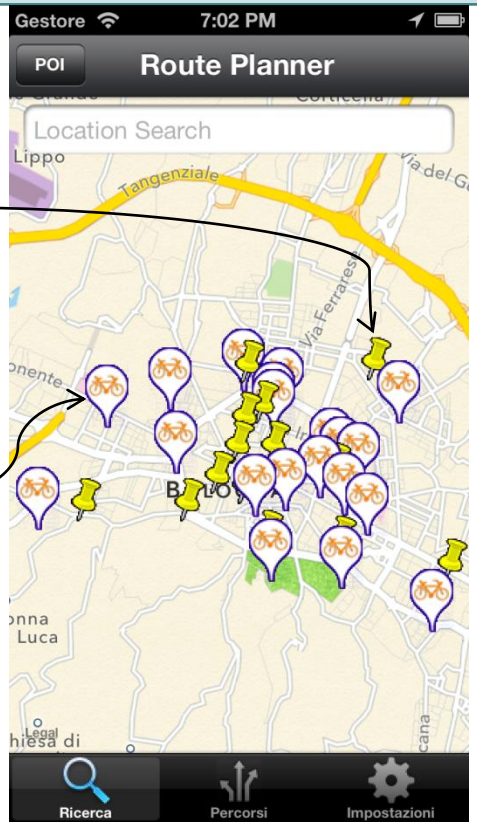
Modena



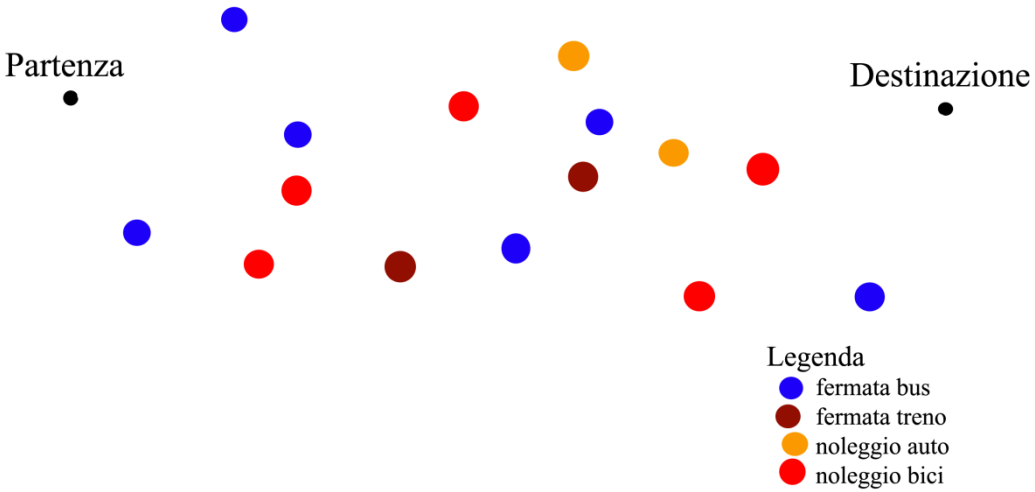
Colonnine di ricarica per i veicoli elettrici

Postazioni di bike sharing

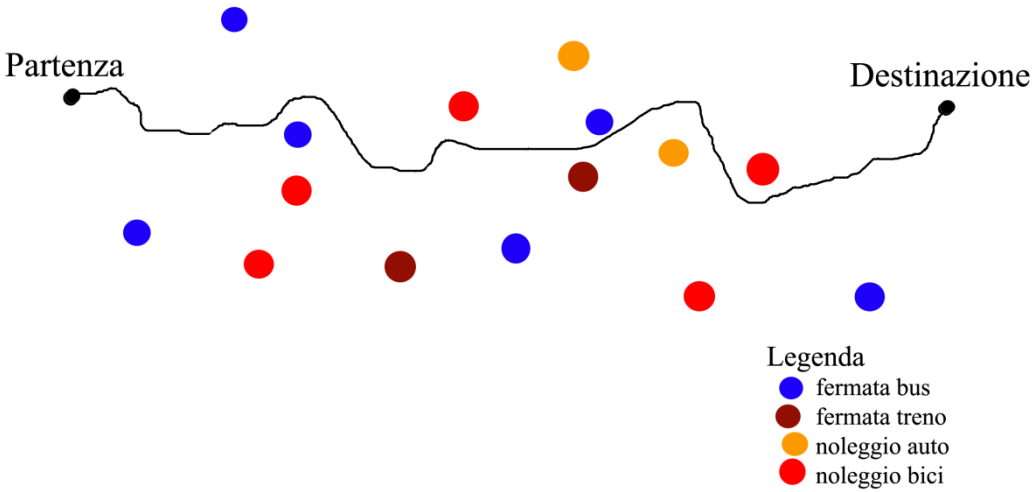
Bologna



Algoritmo sviluppato

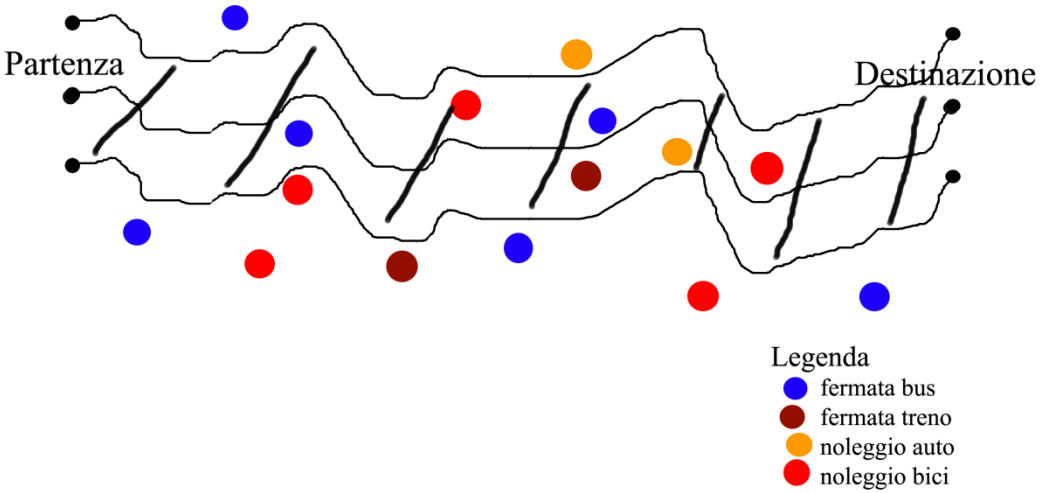


Algoritmo sviluppato



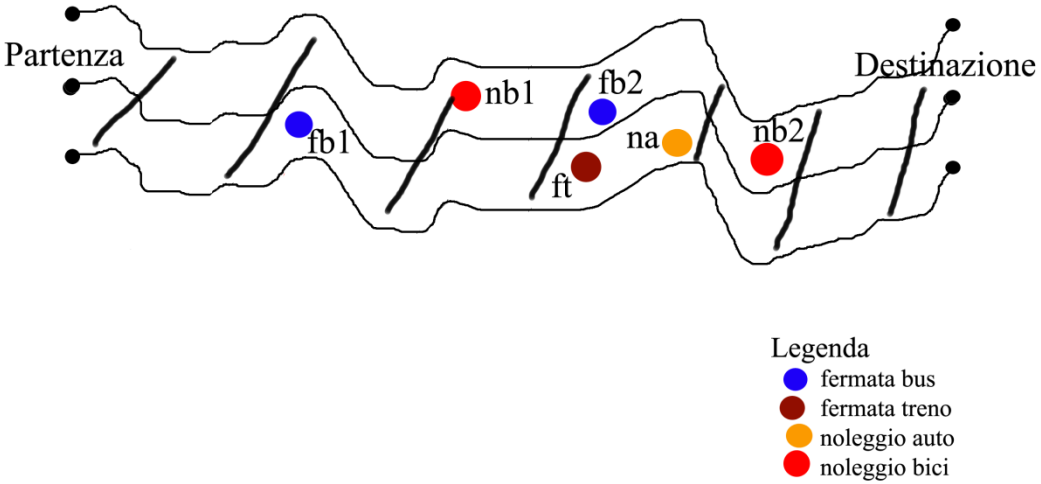
1. Calcolo del percorso eseguito da Google Maps fino alla destinazione per avere un idea del tracciato da seguire

Algoritmo sviluppato



1. Calcolo del percorso eseguito da Google Maps fino alla destinazione per avere un idea del tracciato da seguire
2. Calcolo di un intorno proporzionale alla lunghezza del percorso per identificare i punti di interesse

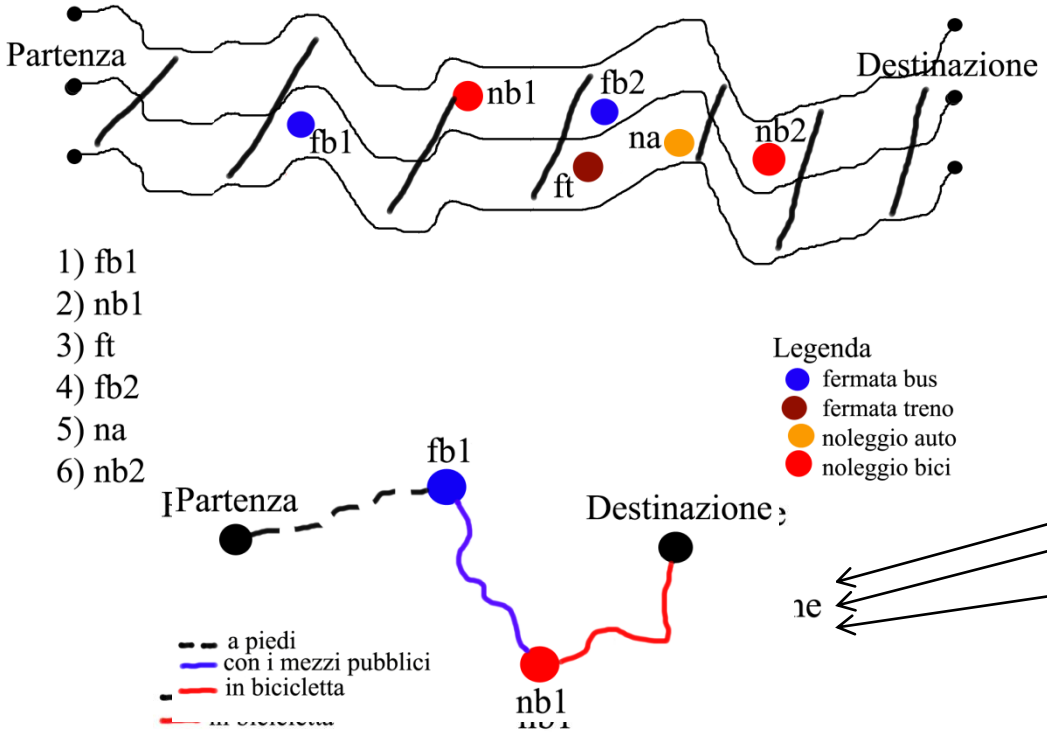
Algoritmo sviluppato



3. Si ordinano in ordine crescente i punti di interesse trovati in base alla distanza dall'origine (dal più vicino al più lontano dall'origine):

a) fermata_bus1 (fb1);
b) noleggio_bici1 (nb1);
c) fermata_treno (ft);
d) fermata_bus2 (fb2);
e) noleggio_auto (na);
f) noleggio_bici2 (nb2)

Algoritmo sviluppato

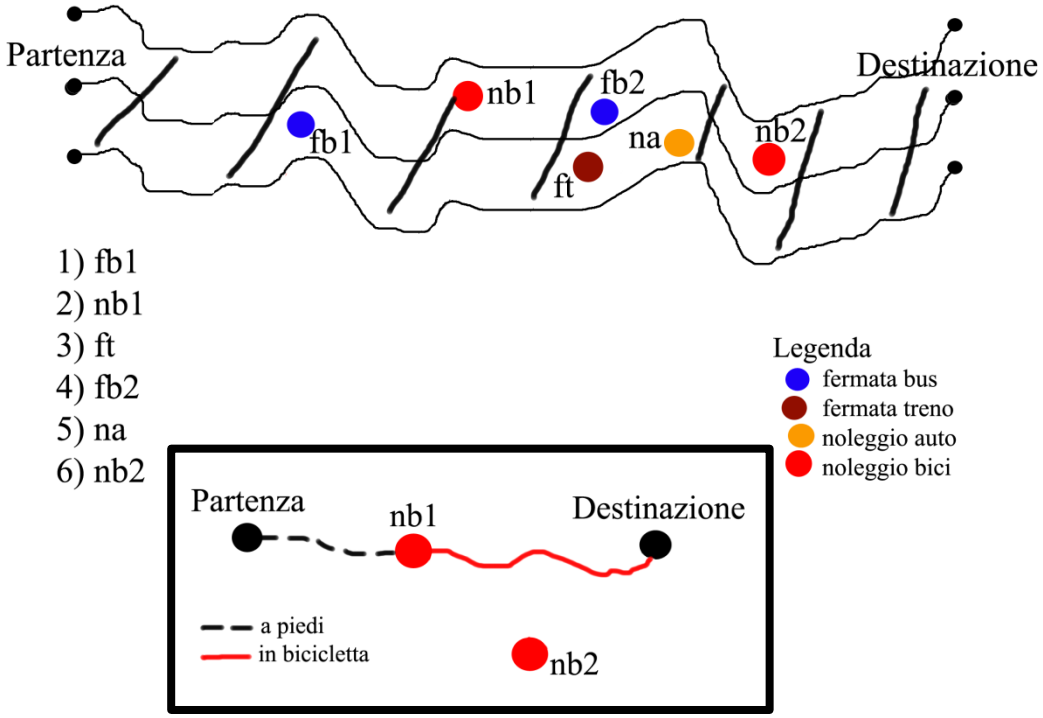


4. Calcolo delle combinazioni semplici $C_{n,1}, C_{n,2}, \dots, C_{n,n}$ dei punti di interesse presi in considerazione. Nell'esempio $n = 6 = n° \, punti \, di \, interesse$. Ogni gruppo di ogni combinazione rappresenterà un percorso.

$C_{6,1} = (fb1), (nb1), (ft), (fb2), (na), (nb2)$

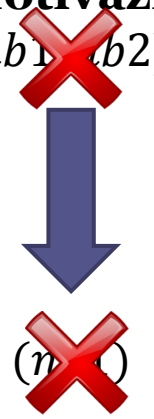
$C_{6,2} = (fb1, nb1), (fb1, ft), (fb1, fb2) \dots$

Algoritmo sviluppato



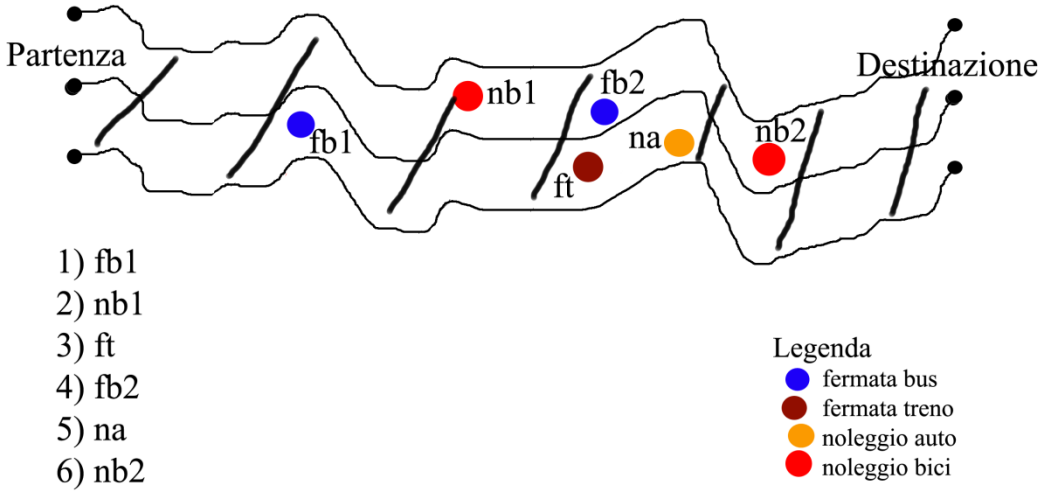
5. Applicazione filtro: eliminazione dei gruppi che presentano due o più mezzi di trasporto uguali adiacenti

Motivazione:



Già calcolato

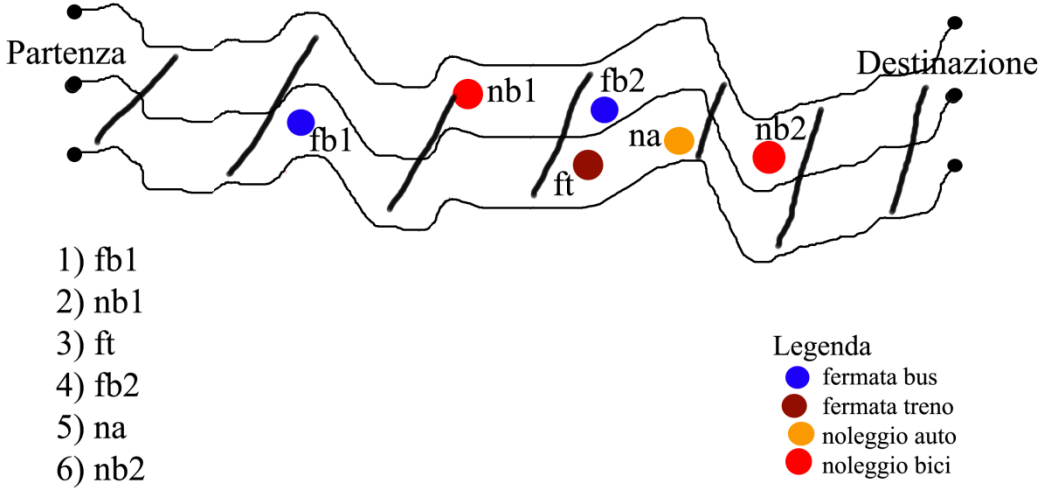
Algoritmo sviluppato



6. Aggiunta dei percorsi che utilizzano un solo mezzo di trasporto dall'origine alla destinazione senza effettuare tappe intermedie.

- Si hanno solo due possibili percorsi dato che si può andare dall'origine alla destinazione soltanto a piedi e con i mezzi pubblici

Algoritmo sviluppato



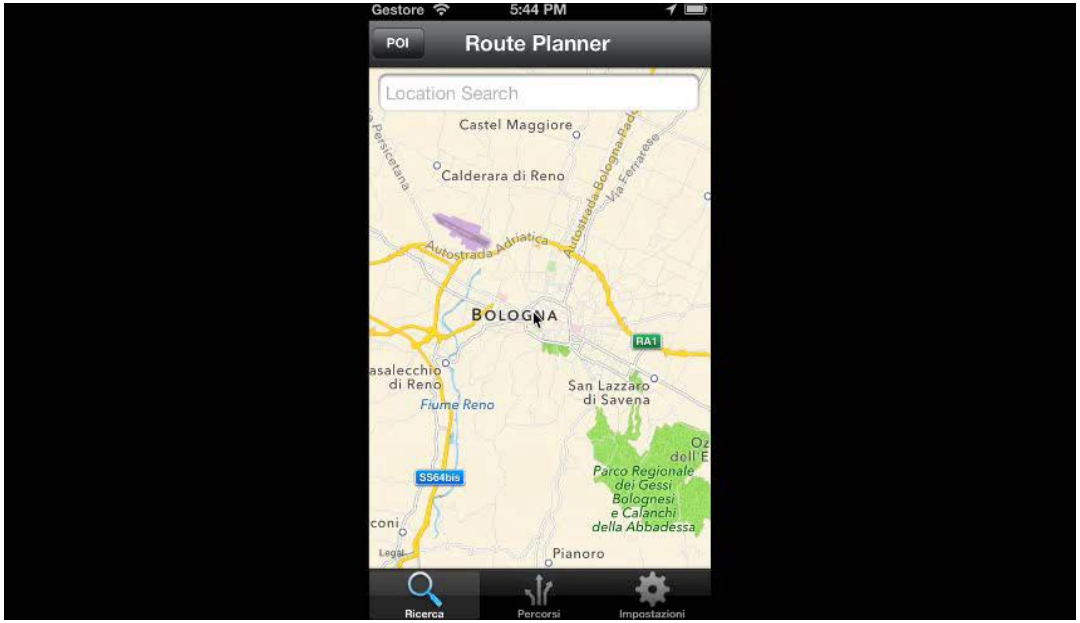
7. Calcolo di tutti i percorsi con Google Maps.

8. I percorsi calcolati vengono ordinati in ordine crescente nei tempi di percorrenza e visualizzati all'utente

Considerazioni algoritmo

- Si suppone che una volta presa la bicicletta in una postazione di bike sharing, sia possibile lasciarla ovunque. Stesso discorso vale per l'automobile
- Google Maps non fornisce percorsi con la bicicletta. Si è quindi calcolato il percorso a piedi tenendo conto di una velocità media di 20 km/h
- Vengono aggiunti dei ritardi nelle tratte percorse in bicicletta e in automobile rispettivamente di 5 e 20 minuti

Esempio utilizzo "Route Planner"



- Esempio di calcolo del percorso da Viale del Risorgimento, 2 – Bologna, sede del DEIS a Via Zamboni, 33 – Bologna, sede centrale dell'università di Bologna

Strumenti utilizzati

- Framework Titanium
- Linguaggio di programmazione JavaScript
- Google Maps API

Titanium Appcelerator

- Framework Cross Platform sviluppato da Appcelerator Inc. ed introdotto nel Dicembre 2008

  Vantaggi:
- Permette di sviluppare un'applicazione con un solo codice JavaScript e di poterla eseguire su più piattaforme hardware (iOS, Android, BlackBerry)
- Compila il codice in un'app nativa per cui si ha una maggiore velocità di esecuzione

Titanium Appcelerator

Svantaggi:

- Essere vincolati ad una specifica piattaforma
- Lentezza di esecuzione nell'accesso alle risorse locali
- Alcuni bug presenti nel framework

Motivazione dell'utilizzo di Titanium

- Grande semplicità nella programmazione

Google Maps API

- Directions API -> 2500 richieste massime in 24h (considerando un singolo dispositivo)
- Places API -> 100.000 richieste massime in 24h (considerando tutti i dispositivi che effettuano le richieste -> presente nella richiesta un'API key condivisa da tutti i dispositivi)
  - Place Search
  - Place Details
  - Place Actions
  - Place Autocomplete

Conclusioni

- Importanza degli Open Data per avviare attività di mercato, prendere decisioni migliori e identificare nuove opportunità
- "Route Planner" possibile grazie alla presenza di Open Data relativi alla mobilità su Google Maps
- "Route Planner" vantaggiosa per i cittadini avendo a disposizione più Open Data

Sviluppi futuri

- Scelta mezzi preferiti
- Salvataggio destinazioni preferite
- Opzione numero di cambi di mezzi di trasporto
- Costo economico del percorso
- Adattamento ad iOS 7

Grazie per l'attenzione