



TPER Bus Finder

Final report

Table of Contents

I.	Scopo dell'applicazione	2
II.	Funzionalità previste	2
III.	UML Diagrams	3
	Use Case Diagram	
	Class Diagram	
	Activity Diagram	
IV.	Caratteristiche dei <i>web services</i> di TPER	5
	Dati sulle linee	
	Dati sulle fermate	
	Dati sulle linee come sequenze di fermate	
	Dati sull'orario di arrivo dei prossimi autobus	
V.	Requisiti	7
VI.	Progettazione e implementazione	7
	Scelte relative all'organizzazione dei dati	
	Scelte relative all'organizzazione delle funzionalità	
VII.	Difficoltà e soluzioni	11
	Procedure asincrone e popolazione di <i>listview</i>	
	<i>Output</i> non strutturato e <i>parsing</i> mediante <i>regular expression</i>	
	Rimozione del contenuto di una tabella preservando i valori di un campo	
VIII.	Criticità riscontrate	12
	Ridondanza dei dati	
	Consistenza linguistica	
	<i>Human-readable</i> versus <i>machine-readable</i>	
	Lentezza	
IX.	Idee per estensioni	13
	Maggiori funzionalità nell'ambito del trasporto su gomma	
	Estendere l'applicazione al trasporto ferroviario	

Scopo dell'applicazione

L'app sfrutta i *web services* e gli *open data* messi a disposizione dall'Azienda dei Trasporti Pubblici dell'Emilia-Romagna (TPER), allo scopo di facilitare la fruizione del servizio da parte degli utenti.

Al momento l'applicazione concerne solo il trasporto su gomma e consente all'utente di trovare le fermate più vicine e di visualizzare gli orari, aggiornati in tempo reale, delle linee di autobus selezionate.

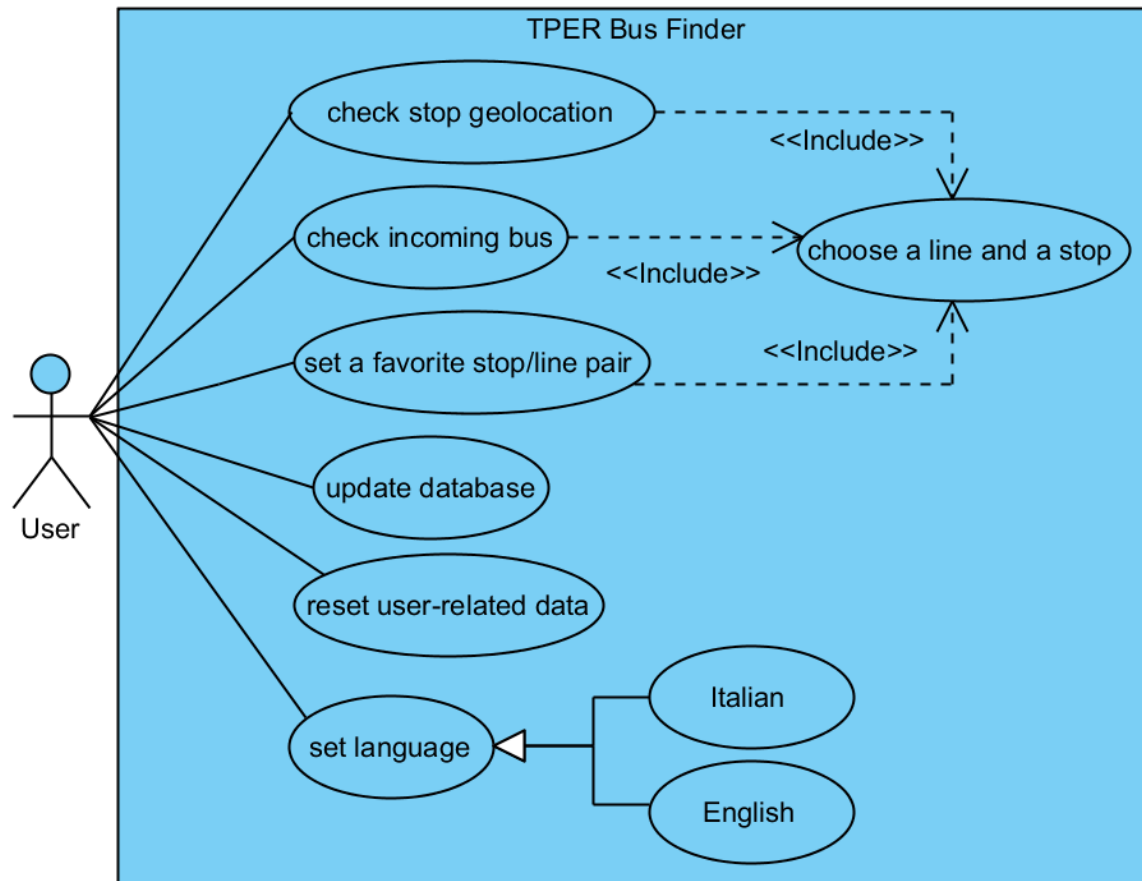
Funzionalità previste

Si prevede la possibilità, da parte dell'utente, di:

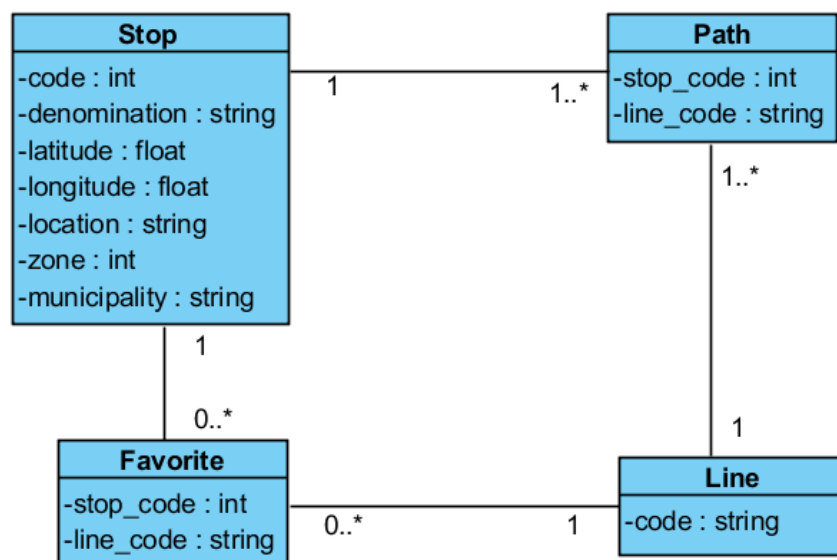
- selezionare una linea di autobus tra tutte le linee attive;
- visualizzare le fermate di una linea di autobus, ordinate per distanza dall'utente (o più precisamente, dal cellulare);
- visualizzare la posizione di una fermata mediante Google Maps;
- ottenere, in tempo reale, gli orari di arrivo dei due autobus più prossimi, relativamente ad una linea e ad una fermata selezionate;
- salvare coppie linea-fermata al fine di rendere il più rapido possibile l'accesso all'informazione dell'orario di arrivo degli autobus;
- aggiornare il *database* e rimuovere i dati generati dall'utente sulle preferenze e sulle linee maggiormente selezionate;
- impostare la lingua impiegata nell'applicazione (inglese o italiano).

UML Diagrams

Use Case Diagram

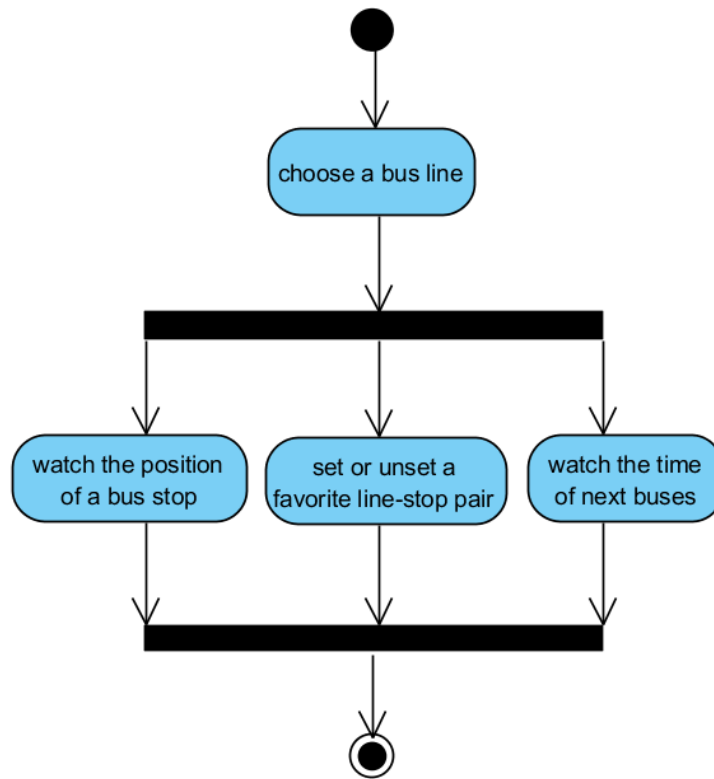


Class Diagram

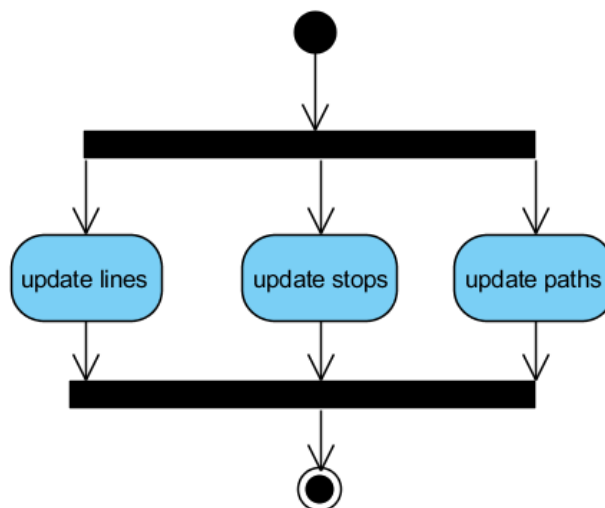


Activity Diagram

- Stops List Activity:



- Update Activity:



Caratteristiche dei web services di TPER

Il servizio di *Open Data* offerto da TPER è descritto formalmente da un *file* di tipo WSDL (*Web Service Definition Language*)¹.

Le *query* di tipo *HTTP Get* impiegate ai fini del progetto sono le seguenti:

Dati sulle linee

URL	https://solweb.tper.it/tperit/webservices/opendata.asmx/OpenDataLinee
Output	<pre><NewDataSet> <Table> <codice_linea>1</codice_linea> </Table> <Table> <codice_linea>101</codice_linea> </Table> ... </NewDataSet></pre>

Dati sulle fermate

URL	https://solweb.tper.it/tperit/webservices/opendata.asmx/OpenDataFermate
Output	<pre><NewDataSet> <Table> <codice>1</codice> <denominazione>STAZIONE CENTRALE</denominazione> <ubicazione>PIAZZA MEDAGLIE D'ORO (PENSILINA C)</ubicazione> <comune>BOLOGNA</comune> <coordinata_x>686344</coordinata_x> <coordinata_y>930918</coordinata_y> <latitudine>44.505736</latitudine> <longitudine>11.343174</longitudine></pre>

¹ <https://solweb.tper.it/tperit/webservices/opendata.asmx?wsdl>

	<pre> <codice_zona>500</codice_zona> </Table>... </NewDataSet> </pre>
--	---

Dati sulle linee come sequenze di fermate

URL	https://solweb.tper.it/tperit/webservices/opendata.asmx/OpenDataLineeFermate
Output	<pre> <NewDataSet> <Table> <codice_linea>1</codice_linea> <codice_fermata>110113</codice_fermata> <denominazione>STAZIONE CENTRALE</denominazione> <ubicazione>PIAZZA MEDAGLIE D'ORO (PENSILINA C)</ubicazione> <comune>BOLOGNA</comune> <coordinata_x>686344</coordinata_x> <coordinata_y>930918</coordinata_y> <latitudine>44.505736</latitudine> <longitudine>11.343174</longitudine> <codice_zona>500</codice_zona> </Table>... </NewDataSet> </pre>

Dati sull'orario di arrivo dei prossimi autobus

URL	https://solweb.tper.it/tperit/webservices/hellobus.asmx/QueryHellobus?fermata=6212&linea=19&oraHHMM=093
Output	<pre> <string xmlns="https://solweb.tper.it/tperit/webservices/hellobus.asmx"> TperHellobus: (x09:30) 19C Previsto 09:34, 19 Previsto 09:41 </string> </pre>

I parametri per quest'ultima interrogazione sono:

- codice della fermata;
- codice della linea;
- orario in formato HHMM.

Requisiti

L'applicazione richiede una connessione attiva ad Internet per l'esecuzione delle *query*. Inoltre richiede la presenza del *GPS receiver* al fine di calcolare la posizione del cellulare e quindi la distanza in metri dalle fermate degli autobus.

Progettazione e implementazione

Scelte relative all'organizzazione dei dati

Tutti i dati (linee, fermate, percorsi, orari) sono soggetti a fluttuazione nel tempo.

L'orario è sicuramente un dato che varia ogni giorno, a causa delle differenze di traffico e quindi velocità media degli autobus. Non viene quindi salvato ma ottenuto *real-time* dai *server* di TPER ad ogni richiesta dell'utente.

Di converso, i dati relativi a linee, fermate e percorsi sono potenzialmente soggetti a mutazione, ma con una frequenza sporadica. Tendenzialmente i percorsi cambiano nel caso di lavori stradali e quindi deviazioni che una particolare linea di autobus può subire. Esistono linee di autobus stagionali nelle zone marittime che sono attive solo durante l'estate. Esistono fermate che vengono soppresse e nuove fermate che vengono create.

Date queste considerazioni si è stabilito di salvare su *database* SQLite i dati soggetti a variazione sporadica (linee, fermate, percorsi). In particolare tenendo presente che l'ottenimento degli stessi richiede un significativo consumo di tempo, in quanto si tratta di eseguire il *download* ed il *parsing* di tre *file* XML, due dei quali di dimensioni non indifferenti (circa 10.000 elementi il primo, circa 20.000 il secondo).

Le tabelle create sono le seguenti:

- Lines:

Name		Type	Primary key
LINE_ID		text	x
LINE_USAGE		integer	

- Stops:

Name	Type	Primary key
STOP_ID	integer	x
STOP_ZONE	integer	
STOP_DENOMINATION	text	
STOP_LOCATION	text	
STOP_MUNICIPALITY	text	
STOP_LATITUDE	real	
STOP_LONGITUDE	real	

- Paths:

Name	Type	Primary key
STOP_ID	text	x
LINE_ID	integer	x

- Favorites:

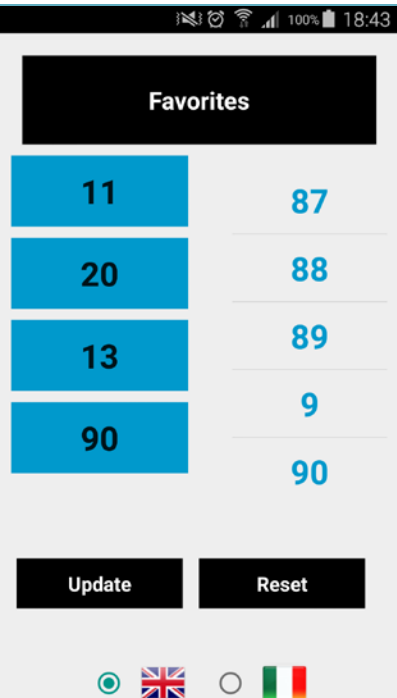
Name	Type	Primary key
STOP_ID	integer	x
LINE_ID	text	x

A questo punto si pone il problema dell'aggiornamento dei dati. Non è stato riscontrato un metodo automatico efficiente per identificare la versione dei dati scaricati. Ad esempio non esiste una *query HTTP Get* finalizzata ad ottenere il *timestamp* del *file* XML scaricato. Di conseguenza, in mancanza della possibilità di un aggiornamento automatico, si è deciso di lasciare la scelta nelle mani dell'utente.

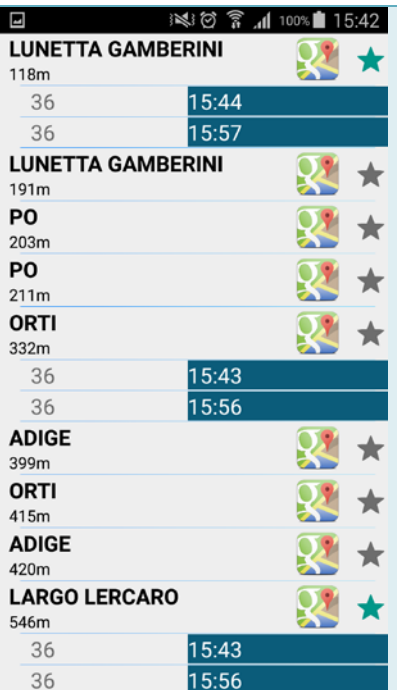
Scelte relative all'organizzazione delle funzionalità

L'applicazione è suddivisa nelle seguenti *activity*:

- Main Activity:

Contents	Screenshot
<ul style="list-style-type: none">• In alto, un pulsante per accedere ai preferiti;• Al centro, due liste: una <i>listview</i> (a destra) contenente tutte le linee attive e una <i>gridview</i> (a sinistra) popolata con le linee precedentemente scelte, ordinate per numero di volte in cui sono state scelte (in ordine decrescente);• Sotto, due pulsanti: uno per aggiornare il <i>database</i> e uno per rimuovere tutti i dati generati dall'utente (i preferiti e le scelte delle linee);• In fondo, due <i>radiobutton</i> contrassegnati da immagini, per modificare la lingua usata nell'applicazione (italiano o inglese).	

- Stops List Activity:

Contents	Screenshot
<ul style="list-style-type: none">• Una <i>expandable listview</i> popolata con tutte le fermate, relative alla linea di autobus selezionata, ordinate in modo crescente per distanza dall'utente.• Ogni <i>item</i> della suddetta lista è composto da: denominazione della fermata; distanza in metri dall'utente; immagine cliccabile per visualizzare la posizione della fermata mediante Google Maps; <i>checkbox</i> stilizzata a stella per salvare o cancellare la coppia linea-fermata tra i preferiti.• La selezione di una fermata genera una <i>sublist</i> che verrà riempita in modalità asincrona dal risultato di una <i>query HTTP Get</i> che ottiene gli orari degli autobus che stanno per giungere presso quella fermata.	

- Favorites Activity:

Contents

- In alto, un pulsante per aggiornare gli orari di tutti i preferiti.
- Sotto, una *listview* contenente i preferiti ordinati in modo alfabetico per nome della linea e per nome della fermata.
- Ogni elemento della lista presenta: la denominazione della linea e della fermata; la distanza in metri dall'utente; l'indicazione degli autobus in imminente arrivo; l'immagine cliccabile per geo-localizzare la fermata; l'immagine cliccabile per aggiornare il singolo elemento; l'immagine cliccabile per eliminare il singolo elemento.

Screenshot

The screenshot shows a mobile application interface for bus favorites. At the top, there is a blue bar with the text "Update all". Below this, the list of favorites is displayed. Each entry includes the line name and number, the stop name, the distance in meters, and a table of arrival times. To the right of the arrival times are three icons: a map pin, a refresh arrow, and a minus sign.

Linea	Fermata	Distanza	Arrivo 1	Arrivo 2
11: DANTE		1748m		
11C			15:47	
11B			15:55	
11: LUNETTA GAMBERINI		118m		
11B			15:43	
11A			15:51	
13: STERLINO		1012m		
13			15:43	
13			15:51	
90: ORTI		332m		
90			15:56	
90			16:26	
90: OSPEDALE BELLARIA				

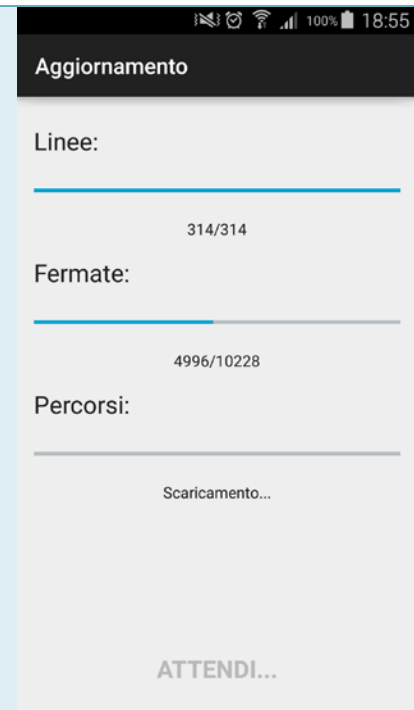
- Map Activity:

Contents	Screenshot
<ul style="list-style-type: none"> • La mappa generata mediante Google Maps con la posizione della fermata selezionata. 	 <p>The screenshot shows a Google Map interface. The map displays a street grid with labels for "Via Giuseppe Ruggi", "Via Giuseppe Dagnini", and "Via Beniamino Gigli". A blue location pin is placed on the map, indicating the selected bus stop. To the right of the pin, the text "Parrocchia S. Severino" is visible. The map includes standard Google Maps controls like a compass, a location icon, and zoom in/out buttons.</p>

- Update Activity:

Contents

- L'indicazione dell'attività di *download* e di *parsing* dei dati afferenti a linee, fermate e percorsi.
- In basso, il pulsante per ritornare nella *main activity*.



Difficoltà e soluzioni

Procedure asincrone e popolazione di *listview*

La maggiore difficoltà è stata riscontrata nella gestione delle interrogazioni *HTTP Get*. A tal fine, si è fatto uso dell'*abstract class AsyncTask*, i cui metodi principali sono: `onPreExecute()`, `doInBackground()`, `onProgressUpdate()`, `onPostExecute()`.

Un aspetto critico è risultato essere la popolazione della *listview* e della *expandable listview* con contenuto ricevuto in modalità asincrona. La soluzione è stata la creazione degli elementi contenitivi (*textView*) riempiti con stringhe vuote e immediatamente dopo l'esecuzione del relativo *AsyncTask*. Nel metodo `onPostExecute()`, ovvero al termine dell'attività asincrona di ricezione ed elaborazione dei dati, viene chiamata la funzione `notifyDataSetChanged()`, per segnalare alla *listview* che la sua *data source* è cambiata e quindi è necessario rigenerare la visualizzazione.

Output non strutturato e parsing mediante regular expression

L'interrogazione *HTTP Get* per l'ottenimento dell'orario di arrivo degli autobus genera una risposta testuale non strutturata. Di conseguenza, si è resa necessaria l'adozione di una *regular expression* finalizzata a decostruire la risposta testuale, identificando il

pattern costitutivo ed estraendo i dati affinché questi possano essere consumati dall'applicazione. La *regular expression* utilizzata è la seguente:

(\\S+) Previsto ([0-2][0-9]:[0-5][0-9])

Questa viene eseguita contro un testo come il seguente:

TperHellobus: (x09:30) 19C Previsto 09:34, 19 Previsto 09:41

Il risultato è l'estrazione dei seguenti valori:

19C	09 : 34	19	09 : 41
-----	---------	----	---------

Rimozione del contenuto di una tabella preservando i valori di un campo

L'aggiornamento dei dati prevede la *tabula rasa* della tabella Lines, la quale contiene anche la colonna (**LINE_USAGE**) relativa alla quantità di volte in cui le linee sono state selezionate da parte dell'utente. Per non perdere il dato, viene salvato in una HashMap, in modo da sfruttare la complessità quasi costante di ricerca tramite funzioni *hash*. Nel momento del *parsing* del file XML, si verifica se la linea estratta dall'elemento XML è una chiave della HashMap, in caso affermativo viene salvato il valore relativo nel campo **LINE_USAGE**, altrimenti questo viene posto a 0.

Criticità riscontrate

Il *web service* attualmente offerto da TPER presenta notevoli criticità. Principalmente:

Ridondanza dei dati

Il *file* XML relativo ai percorsi dovrebbe contenere solo il codice della linea e il codice della fermata e al massimo un codice identificativo del tipo di percorso (ad esempio andata o ritorno), invece ripete i dati sulla singola fermata (denominazione, latitudine, longitudine, etc.) ingigantendo inutilmente la dimensione del *file*.

Consistenza linguistica

Questi *file* XML hanno alcuni *tag* in inglese e altri in italiano. Sarebbe meglio usare solo la lingua inglese.

Human-readable versus *machine-readable*

La risposta all'interrogazione sull'orario di arrivo è in un formato testuale, consono ad una comprensione umana ma inefficace per una lettura automatica. L'ideale sarebbe strutturare meglio i dati, magari passando da XML e JSON che è meno verboso e quindi consente una più rapida comunicazione *client-server*.

Lentezza

L'interrogazione sull'orario di arrivo degli autobus, nonostante la sua semplicità, viene eseguita con un impiego di tempo considerevole (a volte capita di dovere aspettare una decina di secondi). Evidentemente i *server* impiegati da TPER sono poco potenti.

Idee per estensioni

Maggiori funzionalità nell'ambito del trasporto su gomma

Gli *open data* forniti da TPER consentono di realizzare ulteriori funzionalità, maggiormente avanzate. Ad esempio:

- Distinguere effettivamente i percorsi degli autobus: il percorso di andata dal percorso di ritorno.
- Determinare il percorso (sequenza di bus e treni) migliore (secondi diversi parametri, innanzitutto la velocità) per raggiungere un determinato punto, partendo dalla posizione attuale del cellulare.
- Impostare una sveglia connessa al passaggio di un determinato bus o treno, in modo tale che l'orario della sveglia si adatti all'orario effettivo di arrivo del bus o del treno.

Estendere l'applicazione al trasporto ferroviario

TPER si occupa anche di trasporto ferroviario regionale. Quindi è ipotizzabile estendere l'applicazione allo stesso. Le strutture dati e le funzionalità sono le medesime.