

Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

**Sviluppo di un'applicazione mobile
per la valutazione di sostenibilità
della mobilità personale mediante
veicoli elettrici.**

Relatore:
Chiar.mo Prof.
Luciano Bononi

Presentata da:
Gianluca Pasqualicchio

Correlatori:
Dott. Luca Bedogni
Dott. Marco Di Felice

Sessione II
Anno Accademico 2013/2014

Ai miei genitori...

Abstract

La sempre minore disponibilità di combustibili fossili e il crescente inquinamento ambientale, hanno incentivato non solo la ricerca di fonti alternative di energia, ma anche lo sviluppo di nuove tecnologie che ci permettessero di sfruttarle.

Non solo sono stati sviluppati pannelli solari e paleoliche, che ci permettono di ottenere energia rinnovabile; ma anche i "comuni" strumenti sono stati modificati per far fronte a queste esigenze. Tra le più importanti innovazioni, che mirano a rendere più eco-sostenibile la mobilità veicolare, lo sviluppo delle auto elettriche è quella che ha riscosso più interesse.

Nonostante gli ingenti investimenti iniziali, l'immissione nel mercato non sta riscuotendo il successo atteso.

Uno dei maggiori limiti riscontrati è quello che viene definito *Range Anxiety*, ovvero la paura che il veicolo elettrico (EV) non abbia abbastanza autonomia per garantire al conducente il raggiungimento della meta. Inoltre il tutto viene amplificato dai lunghi tempi di ricarica delle batterie e l'elevato costo degli EV.

Per arginare questi problemi, in questo documento viene descritto lo sviluppo di un applicazione Android (WhatIfApp - WIA) per la valutazione della sostenibilità di un viaggio, facendo visualizzare all'utente i tempi di percorrenza (ottimizzati) e i consumi, che un EV dovrebbe affrontare in determinati percorsi.

In particolare, ne viene spiegato l'utilizzo, le funzionalità, le scelte implementative e le future estensioni.

Indice

1	Introduzione	1
1.1	Perchè le vendite sono state così scarse?	1
1.2	WhatIfApp e il <i>range anxiety</i>	2
1.3	Servizi esterni	2
1.3.1	Google Maps Android API	3
1.3.2	Google Maps API Web Services	3
1.3.3	Servizio di route planning	4
2	Stato dell'arte	1
2.1	La mobilità elettrica e UniBo	1
2.1.1	Internet of Energy - (IoE)	1
2.2	Lavori simili	4
2.2.1	iEV	4
3	Android	7
3.1	Introduzione ad Android	7
3.2	Storia	8
3.3	Perchè Android?	9
3.4	Sviluppo di un applicazione	10
3.4.1	Creazione, compilazione, emulazione	10
3.4.2	Elementi principali di un app Android citati su WIA	11
4	WhatIfApp - (WIA)	17
4.1	Tipologie di tratte	17

4.2	Creazione di un Offline TRIP	18
4.2.1	Inserimento delle locazioni (Partenza, Destinazione, Intermezzi)	19
4.2.2	Tracciare il percorso	21
4.2.3	Modifiche alla tratta	22
4.2.4	Salvataggio della tratta	22
4.3	Creazione di un Recorded TRIP	23
4.3.1	Registrazione della tratta	24
4.3.2	Salvataggio della tratta registrata	26
4.4	Creazione di un Journey	26
4.4.1	Linea temporale	27
4.4.2	Gestione del Journey	28
4.5	Modifica/Elimina di un trip/journey	30
4.6	Simulazione di un Trip/Journey	31
4.6.1	Simulazione di un Trip	32
4.6.2	Simulazione di un Journey	35
	Conclusioni	39
	Riferimenti Bibliografici	45

Capitolo 1

Introduzione

Ormai da alcuni anni, diverse case automobilistiche, hanno cominciato ad introdurre nel mercato autoveicoli elettrici per tentare di porre rimedio al problema dell'eccessivo consumo di combustibili fossili nel mondo e per ridurre l'inquinamento ambientale, soprattutto nelle grandi città.

Nonostante l'immissione nel mercato di questi veicoli fu accolto con feedback positivi e numerosi pre-ordini; gli effettivi acquisti delle auto elettriche sono stati decisamente troppo pochi rispetto alle aspettative.

1.1 Perché le vendite sono state così scarse?

Sono diversi i problemi che attualmente contribuiscono a limitare gli acquisti di veicoli elettrici da parte degli utenti:

- **Costo:** la differenza di prezzo tra un autoveicolo elettrico e uno a benzina/diesel dello stesso modello è quasi il doppio, per lo più a causa dell'elevato costo della batteria.
- **Lunghi periodi di ricarica:** ricaricare la batteria richiede, spesso, diverse ore di sosta presso il punto di ricarica. Risolvibile, comunque, tramite servizi di *Battery sharing*, ovvero la sostitu-

zione di una batteria scarica con una carica e con lo sviluppo di colonnine con una più elevata potenza di carica; anche se, ad una più elevata velocità di ricarica, corrisponde un più veloce deterioramento.

- **L'esiguo numero dei servizi di ricarica:** nonostante nelle grandi città, il numero delle stazioni di ricarica risulta adeguato, il problema è accentuato al di fuori di esse, anche se il numero di stazioni sta aumentando velocemente.
- **Range anxiety:** ovvero la paura che il veicolo elettrico non abbia abbastanza autonomia (range) per arrivare a destinazione.

C'è anche da dire, che un numero sempre maggiore di finanziamenti viene stanziato per cercare di arginare il più possibile questi problemi e per garantire un servizio sempre più efficiente e competitivo. Il risultato è che, nonostante la quantità ancora bassa di veicoli elettrici venduti, il numero di acquisti effettuati ogni anno è in netta crescita; perciò vendite basse sì, ma c'è anche tanto ottimismo.

1.2 WhatIfApp e il *range anxiety*

L'idea che sta alla base di WhatIfApp è quella di dare agli utenti un'applicazione per smartphone (in particolare Android), comoda, semplice e facile all'uso, che permetta all'utente di farsi un'idea su quali sono i consumi che un'auto elettrica dovrà affrontare in determinate tratte e dove eventualmente dovrà andare a ricaricarsi. Utile ad esempio in previsione di un acquisto.

1.3 Servizi esterni

Per la realizzazione dell'applicazione sono stati utilizzati alcuni servizi esterni, tra i quali:

- **Google Maps Android API**
- **Google Maps API Web Service**
- **Un servizio web di route planning per scenari di mobilità elettrica**

1.3.1 Google Maps Android API

Al fine di rendere più interattive e intuitive la creazione delle tratte, la loro modifica e per dare una visione su mappa dei dati raccolti durante la simulazione, si è scelto l'uso delle mappe fornite da Google e alcune tra le API messe da loro a disposizione. Grazie alle Google Maps API per Android[1], è possibile aggiungere nella propria applicazione mappe basate sui dati Google Maps. Le API automaticamente gestiscono l'accesso al server delle Google Maps, il download dei dati, la visualizzazione della mappa e le risposte alle varie gesture. Le API possono essere anche utilizzate per l'aggiunta di Markers e Poligoni per rendere il tutto più *user-friendly*.

1.3.2 Google Maps API Web Services

Altre API utilizzate per l'applicazione sono:

- Directions API[2]: Servizio chiamato con una richiesta HTTP utilizzato per la restituzione dei percorsi tra varie locazioni. Tramite questo servizio è possibile la restituzione di un percorso per diversi tipi di trasporto (autoveicoli, a piedi, in bici, ecc...). La richiesta viene fatta specificando Partenza e Destinazione (anche possibili waypoints) tramite stringa (Es. 'Bologna, BO, Italy') o come coordinate geografiche nella forma 'latitudine,longitudine'.
- Geocoding API: Servizio che permette il Geocoding, ovvero permette la conversione di indirizzi (Es. 'Via Irnerio 33, Bologna, BO')

in coordinate geografiche nella forma 'latitudine,longitudine', utile soprattutto per il piazzamento dei vari marker durante la creazione di tratte semplice; e il Reverse geocoding, ovvero il servizio opposto, converte coordinate geografiche in indirizzi.

Entrambe le API forniscono risultati in Json o XML (per l'applicazione si è scelto la restituzione in JSON).

Entrambi i servizi 'soffrono' di due limitazioni, ovvero di un numero di richieste per secondo e di richieste giornaliere limitate, diverso in base al tipo di utilizzo che si fa delle API, free o business.

Per come è stata strutturata la WhatIfApp, il limite di richieste al secondo non dovrebbe rappresentare un problema; mentre, il limite di richieste giornaliere, ne impedisce un utilizzo intensivo.

1.3.3 Servizio di route planning

Per la pianificazione del percorso con auto elettriche in determinate tratte, viene utilizzato un servizio web di route planning [4]{**vedi Cap. 2.1**}

Capitolo 2

Stato dell'arte

2.1 La mobilità elettrica e UniBo

L'università di Bologna partecipa attivamente allo sviluppo di servizi a favore della mobilità elettrica ed eco-sostenibile. Uno degli ultimi progetti a cui ha dato un contributo e da poco concluso è Internet of Energy.

2.1.1 Internet of Energy - (IoE)

Intenet of Energy([3]) è un progetto di ricerca europeo, mantenuto da ARTEMIS (Advanced Research and Technology for Embedded Intelligence and Systems) a cui hanno partecipato decine di partner da tutta europa. Lo scopo del progetto è stato quello di sviluppare hardware, software e middleware per garantire connettività sicura e interoperabilità tramite connessione con la Smart Grid, al fine di creare un'infrastruttura di mobilità elettrica. La seguente immagine rappresenta l'architettura del progetto mostrandone i principali attori.

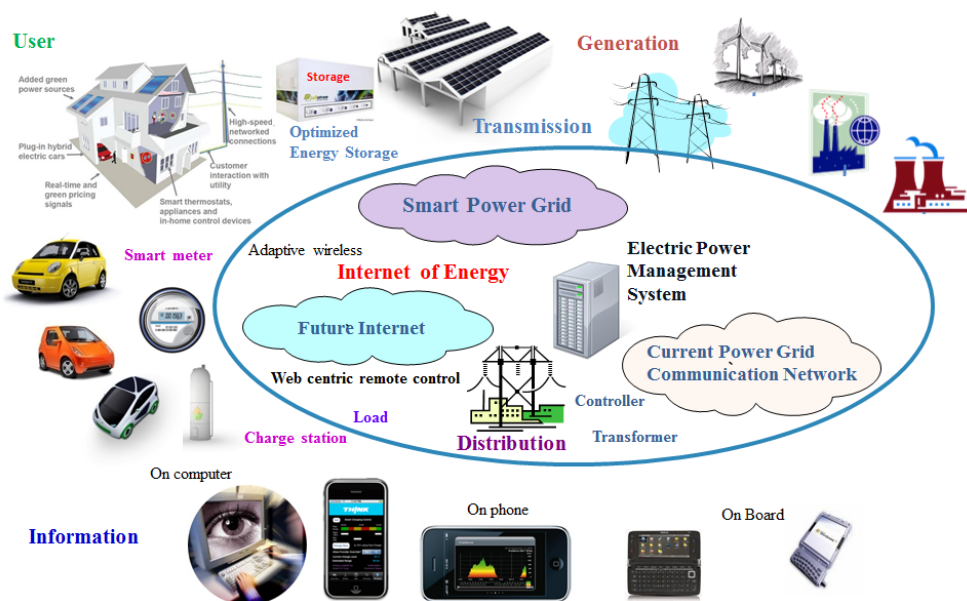


Figura 2.1: Architettura di Internet of Energy

- **Smart-Grid:** una moderna rete di distribuzione elettrica, che tramite l'utilizzo delle più moderne tecnologie di comunicazione, garantisce maggiore efficienza e affidabilità nelle operazioni di conservazione e distribuzione dell'energia, tra cui l'infrastruttura dedicata alla ricarica dei veicoli elettrici.
- **L'utenza:** che interagisce tramite la Smart Grid con i diversi servizi offerti, tra cui, applicazioni mobili o interne ai veicoli.
- **Internet:** rete di comunicazione tra la Smart Grid, i veicoli e l'utenza, che permette l'ottimizzazione delle operazioni di ricarica delle batterie, rispettando i vincoli imposti dal traffico cittadino e dal rapporto tra consumo e disponibilità energetica.

Alcuni tra i contributi che l'Università di Bologna ha portato ad IoE, sono:

- **City Service:** Il City-Service è un'architettura software atta a garantire la comunicazione tra i veicoli elettrici e la Smart-Grid.

In particolare, tramite un'applicazione mobile[5] è possibile effettuare delle richieste di prenotazione di ricarica, che, una volta ricevute dal City-Service, verranno esaminate fornendo, in seguito, la lista delle stazioni di ricarica disponibili.

L'applicazione si occupa anche di fornire all'utente alcuni tra i dati utili riguardo il proprio veicolo, in modo tale che l'utente sappia, in qualsiasi momento, se è necessaria, o meno, una ricarica; ad esempio: lo stato di carica (SOC), il peso, ecc

- **Simulatore:** una piattaforma di simulazione per la visualizzazione del comportamento degli EV posti all'interno di uno scenario urbano (Es. città di Bologna). La simulazione tiene conto dei consumi delle batterie di ogni singola auto, gestendone le richieste di ricarica.

Molto utile nello studio dell'impatto che, l'introduzione degli EV, potrà portare al traffico cittadino.

Tra i lavori svolti dall'Università di Bologna all'interno di IoE è presente anche il web service utilizzato per la simulazione nella WhatIfApp.

Servizio web di Route Planning

L'architettura del web service([4]), utilizzato per la simulazione dei veicoli elettrici sulle varie tratte, è costituita principalmente da tre componenti fondamentali.

- **Consumption model:** fornisce un'approssimazione del consumo, (in kWh) che l'auto elettrica dovrà affrontare nel tratto di strada preso in esame, in base alle caratteristiche dell'auto (es: massa, area della massima sezione trasversale, ecc...).
- **Path finder:** modulo che si occupa del reperimento di informazioni sui percorsi stradali, tra cui i dati altimetrici.

Usato in sinergia con il consumption model per il calcolo del consumo.

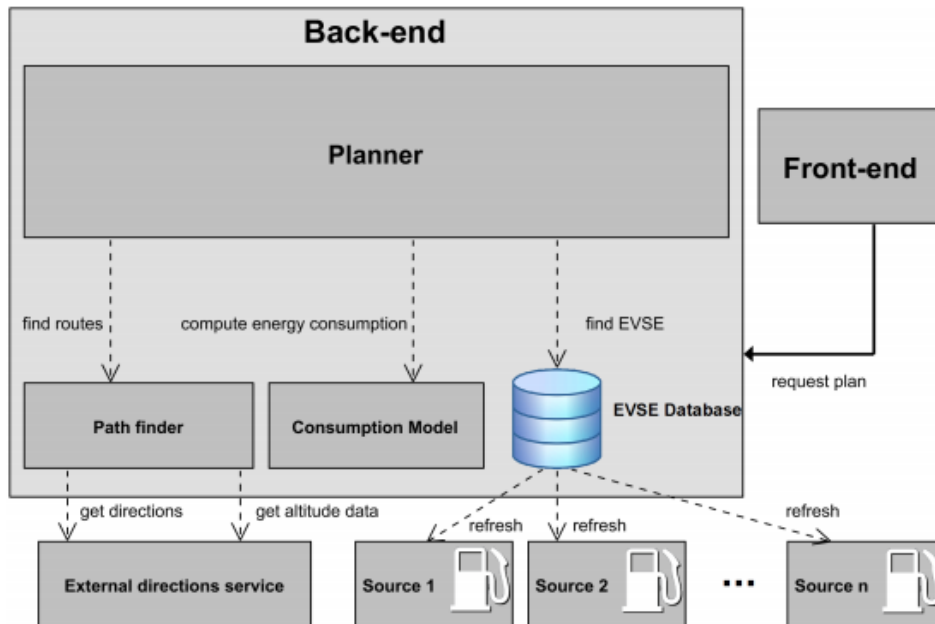


Figura 2.2: Architettura del web service per il route planning

- **Planner:** il cui compito è quello di costruire un grafo dei percorsi possibili, tra la Partenza e la Destinazione, e di selezionare, tra essi, quello più performante, in base ad alcuni parametri (Es. la carica iniziale della batteria e la soglia minima che la batteria dovrà avere al termine della tratta).

2.2 Lavori simili

WhatIfApp nasce dall'idea di sviluppare un applicazione Android come evoluzione di un app già esistente sul mercato, iEV.

2.2.1 iEV

iEV registra dati durante i viaggi degli utenti, per poi analizzarli e simulare il comportamento di una macchina elettrica sullo stesso percor-



Figura 2.3: Icona di iEV

so. Secondo gli sviluppatori di iEV dopo aver compiuto alcuni viaggi per un minimo di quattro settimane, l'utente, in base alle proprie esigenze, dovrebbe essere riuscito a farsi un'idea precisa riguardo all'idoneità dell'auto elettrica su cui si sono fatte le simulazioni.

Come utilizzare iEV

L'applicazione deve essere attivata prima di ogni viaggio ed è stata sviluppata per essere utilizzata all'interno di automobili. Attivare l'applicazione all'interno di un'abitazione non produrrà alcun risultato. Dopo aver terminato un percorso in auto, la registrazione GPS dovrà essere disattivata premendo il pulsante rosso FINE. L'utente sarà in grado di salvare i dati registrati e includerli all'interno dell'analisi di mobilità. I viaggi validi saranno successivamente archiviati e mantenuti in sincronizzazione mediate i server di backup. Possono essere analizzati i dati dei singoli viaggi così come quelli del numero complessivo di viaggi (e.g. distanza di guida, velocità, dispendio di energia, etc.).

Limitazioni

I limiti posti da iEV sono evidenti:

- **La registrazione dei viaggi:** essendo l'unico modo per raccogliere i dati, ne rende poco pratico l'utilizzo; non solo perchè per

avere dei risultati soddisfacenti l'utente debba aspettare, addirittura, un mese, ma anche perchè l'utente è tenuto a ricordarsi sia dell'avvio della registrazione e sia, soprattutto, della fine. Basterebbe quindi dimenticarsi o dell'avvio o della fine del viaggio, per non poter più usufruire dei potenziali dati.

- **Un EV alla volta:** Una volta raccolti i dati, la simulazione vale solo per un EV selezionato prima della registrazione. Questo implica che, per verificare l'idoneità su un veicolo elettrico differente, sarà necessario registrare nuovamente i dati.

Capitolo 3

Android

Il sistema operativo per cui è stata sviluppata l'applicazione è Android; perciò, prima di parlare a fondo della WhatIfApp, sarebbe opportuno prima descrivere alcune strutture e nomenclature utilizzate.



Figura 3.1: Logo Android

3.1 Introduzione ad Android

Android([6]) è un sistema operativo per dispositivi mobili sviluppato da Google Inc. sulla base del kernel Linux e progettato principalmente

per smartphone e Tablet, con interfacce utente specializzate per televisori (Android TV), automobili (Android Auto), orologi da polso (Android Wear), occhiali (Google Glass), e altri.

Android è per la quasi totalità Free and Open Source, ed è rilasciato sotto i termini della licenza libera Apache 2.0 che ne consente la libera modifica e distribuzione del codice sorgente.

3.2 Storia

Nell'ottobre 2003 Andy Rubin, Rich Miner, Nick Sears e Chris White, fondarono una società, la Android Inc. per lo sviluppo di quello che Rubin definì «...dispositivi cellulari più consapevoli della posizione e delle preferenze del loro proprietario». Da ciò probabilmente scaturì la scelta del nome Android.

Inizialmente la società operò di nascosto, rivelando solo di progettare software per dispositivi mobili.

Il 17 agosto 2005 Google ha acquisito l'azienda, in vista del fatto che la società di Mountain View desiderava entrare nel mercato della telefonia mobile. È in questi anni che il team di Rubin comincia a sviluppare un sistema operativo per dispositivi mobili basato sul kernel Linux.

La presentazione ufficiale del robottino verde avvenne il 5 novembre 2007 dalla neonata OHA (Open Handset Alliance), un consorzio di aziende del settore Hi Tech che include Google, produttori di smartphone come HTC e Samsung, operatori di telefonia mobile come Sprint Nextel e T-Mobile, e produttori di microprocessori come Qualcomm e Texas Instruments Incorporated. Il primo dispositivo equipaggiato con Android che venne lanciato sul mercato fu l'HTC Dream, il 22 ottobre del 2008. Da allora, gli aggiornamenti per Android, atti all'aggiunta di funzionalità, migliorie alle prestazioni e all'eliminazione di bug presente in versioni precedenti, sono stati numerosi.

Ogni aggiornamento, a partire dalla versione 1.5, ha assunto nomi di

versi di dolci.

<i>Versione</i>	<i>Data pubblicazione</i>	<i>API level</i>
1.0	23 Settembre 2008	1
1.1	9 Febbraio 2009	2
1.5 <i>Cupcake</i>	30 Aprile 2009	3
1.6 <i>Donut</i>	15 Settembre 2009	4
2.0 <i>Eclair</i>	26 Ottobre 2009	5
2.2 <i>Froyo</i>	20 Maggio 2010	8
2.3 <i>Gingerbread</i>	6 Dicembre 2010	9
3.0 <i>Honeycomb</i>	22 Febbraio 2011	11
4.0 <i>Ice Cream Sandwich</i>	19 Ottobre 2011	14
4.1 <i>Jelly Bean</i>	9 Luglio 2012	16
4.4 <i>Kit Kat</i>	31 Ottobre 2013	19
5.0 <i>Lollipop</i>	3 Novembre 2014	21

Tabella 3.1: Elenco versioni di Android

3.3 Perché Android?

Il motivo principale del perché l'applicazione è stata sviluppata, almeno per adesso, solo per Android, è principalmente dovuto al fatto che il sistema operativo di Google attualmente domina in maniera indiscussa il mercato degli smartphone, almeno per quanto riguarda la quantità di dispositivi venduti.

Infatti secondo ABI Research, nei primi mesi del 2014 sono stati venduti all'incirca 427 milioni di dispositivi di telefonia, di cui 300 milioni sono smartphone; e tra i 427 milioni di dispositivi, circa il 57% utilizza il SO Android.[7]

Facile intuire quindi che, sviluppando APP su Android, si ha automaticamente l'accesso ad un utenza molto più numerosa rispetto agli altri

Sistemi Operativi; inoltre Android non pretende, come ad esempio la Apple, abbonamenti annuali a pagamento per il permesso di sviluppo di applicazioni sul proprio sistema operativo, ma solo se si decide di farne l'upload sul market ufficiale, Google Play. Ad un prezzo comunque accessibile.

3.4 Sviluppo di un applicazione

Le applicazioni Android, tipicamente, vengono sviluppate all'interno di un framework, ovvero un tipo di struttura specifica, molto chiara se per la creazione dell'App ci si affida ad un ambiente di sviluppo (Android SDK) su Eclipse (usato per lo sviluppo di WIA).

Le applicazioni Android sono composte da parti dinamiche, scritte in Java (un suo 'dialetto'), e di parti statiche, scritte in XML.

Le parti statiche sono quelle caratteristiche che non cambiano durante l'esecuzione dell'applicazione, come ad esempio la parte grafica; le parti dinamiche, invece, sono invece gli aspetti dell'applicazione programmatici, come la gestione degli eventi.

3.4.1 Creazione, compilazione, emulazione

L'applicazione, tramite l'SDK, può essere testata sia in emulazione con un dispositivo virtuale e sia un dispositivo reale collegato al device. Prima di eseguire l'applicazione, al dispositivo, tramite il file Manifest.xml, viene descritta l'applicazione, in modo tale da accertarsi che il dispositivo sia in grado di far girare l'app.

Accertata l'idoneità del dispositivo nel far girare l'applicazione, il codice Java e XML vengono compilati per generare un file con estensione .APK, che contiene il bytecode per la cosiddetta Dalvik Virtual Machine (DVM).

Successivamente si passerà all'installazione dell'applicazione sul dispositivo.

Dalvik Virtual Machine

Grazie all'utilizzo dell'SDK è possibile trasformare la nostra applicazione, formata quindi da Java, XML e il Manifest.xml, in bytecode (linguaggio intermedio più astratto tra il linguaggio macchina e il linguaggio di programmazione).

Apparentemente potrebbe sembrare la stessa cosa che succede abitualmente in Java con la Virtual Machine (VM), ma c'è una piccola differenza negli ambienti Android, ovvero che la VM utilizzata è stata riscritta e chiamata Dalvik Virtual Machine (DVM), installato su ogni dispositivo Android, ed essendo uguale per tutti, l'applicazione sviluppata per android potrà essere eseguita su ogni tipo di terminale, indipendentemente dal costruttore e dall'implementazione.

Il risultato positivo della DVM è che quindi molti costruttori di smartphone, scelgono quindi l'ambiente Android sapendo che in questo modo potranno fornire ai propri utenti un ambiente condiviso da moltissimi altri utenti.

3.4.2 Elementi principali di un app Android citati su WIA

Activity

Una Activity è un componente principale di una applicazione Android che fornisce una schermata con cui gli utenti possono interagire per fare una qualsiasi azione: scrivere messaggi, scattare fotografie, ecc...

Una Activity è posizionata di solito su una finestra, che tipicamente occupa tutta la grandezza dello schermo, ma può anche riempire uno

spazio minore o 'galleggiare' sopra altre finestre.

Un'applicazione è composta, solitamente, da diverse Activity legate tra loro. Tipicamente c'è una Activity principale che viene avviata all'apertura dell'applicazione e che serve principalmente come presentazione; successivamente ogni Activity può avviare altre Activity per l'esecuzione di altri compiti.

Ogni volta che viene eseguita una nuova Activity, l'Activity precedente viene messa in stop, e messa dal sistema in una pila (la 'back stack') e quella nuova viene messa in cima allo stack, che avrà quindi il focus, ovvero sarà quella visualizzata sullo schermo. Il 'back stack' segue la politica LIFO (Last In First Out), perciò quando un utente premerà il Back Button, verrà mostrata su schermo l'ultima activity messa in PAUSA.

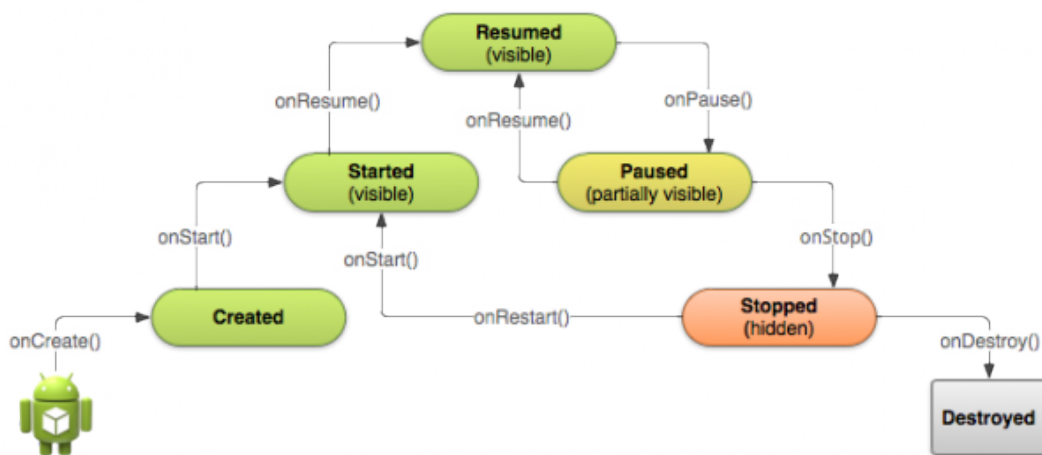


Figura 3.2: Il ciclo di vita di un'activity in Android

In definitiva, i principali stati di un Activity sono:

- **ACTIVE** : L'activity è in cima allo stack, visibile e ha il focus. È quella che riceve gli eventi da parte dell'utente.

- **PAUSED** : Si riferisce al caso di Activity non attive, ma ancora visibili per la trasparenza di quelle superiori o perchè queste non occupano tutto lo spazio a disposizione. Essa non è quindi sensibile agli eventi da parte degli utenti e viene eliminata solamente in situazioni di estrema necessità.
- **STOPPED** : Si riferisce al caso di Activity non attive, nè visibili. Non è ovviamente sensibile agli eventi dell'utente ed è tra le prime candidate a essere eliminata.
- **INACTIVE** : Una Activity si trova in questo stato quando viene eliminata oppure prima di essere creata.

Ogni cambio di stato viene preceduto da una chiamata di una funzione diversa; ad esempio, quando un activity viene chiamata per la prima volta, viene eseguita la funzione `onCreate`, o quando ad un'Activity viene ripassato il focus, viene chiamata la funzione `onResume()` (vedi **fig.3.2**).

Geocoder

La classe `geocoder`[9] permette il geocoding e il reverse geocoding. Il *geocoding* è il processo di trasformazione di un indirizzo o una qualsivoglia descrizione di una locazione in coordinate geografiche (latitudine, longitudine).

Il *reverse geocoding* è, al contrario, il processo di trasformazione di coordinate geografiche (latitudine, longitudine) in un indirizzo.

La quantità di dettagli restituiti durante il processo di reverse geocoding può variare, ad esempio potrà contenere l'indirizzo del luogo conosciuto più vicino o può essere restituito solo ad esempio il nome della città o il codice postale.

Il Geocoder richiede un servizio di backend che non è incluso nel core del framework Android.

Senza la presenza del suddetto servizio, tutte le chiamate del Geocoder restituiranno una lista vuota.

Events

Uno dei concetti principali nell'utilizzo dei componenti di un'interfaccia grafica riguarda la gestione degli eventi. I due più citati eventi utilizzati in WIA sono:

- **Click:** Evento di selezione di un componente, ad esempio un Button o un elemento qualsiasi nella Activity.
- **Long Click:** Evento di hold, ovvero di selezione prolungata di un componente.

Ovviamente esistono molti altri eventi, non chiamati in WIA, quali:

- **Focus Change:** Evento di acquisizione o perdita del focus da parte di un componente.
- **Key:** Evento di selezione di un tasto.
- **Touch:** Evento di Touch.
- **CreateContextMenu:** Evento di creazione del menu contestuale.

Dialogs

Un Dialog è una piccola finestra (popup) che, tipicamente, serve per chiedere conferma prima di procedere con determinate azioni o per l'inserimento di informazioni aggiuntive.

Widget

Qui di seguito sono elencati e brevemente descritti tutti i vari widget ([11]) utilizzati da WIA:

- **Button:** rappresenta semplicemente un bottone. Un *Button* può essere premuto o cliccato per compiere una qualsiasi azione.
- **CheckBox:** particolare estensione di *Button* che ci permette di sapere se un particolare elemento è selezionato o meno.
- **EditText:** la quasi totalità delle caratteristiche di una *EditText* sono già contenute all'interno della sua classe parent, ovvero *TextView*. La differenza sostanziale è quindi rappresentata dalla possibilità di modificare il testo semplicemente toccando il widget.

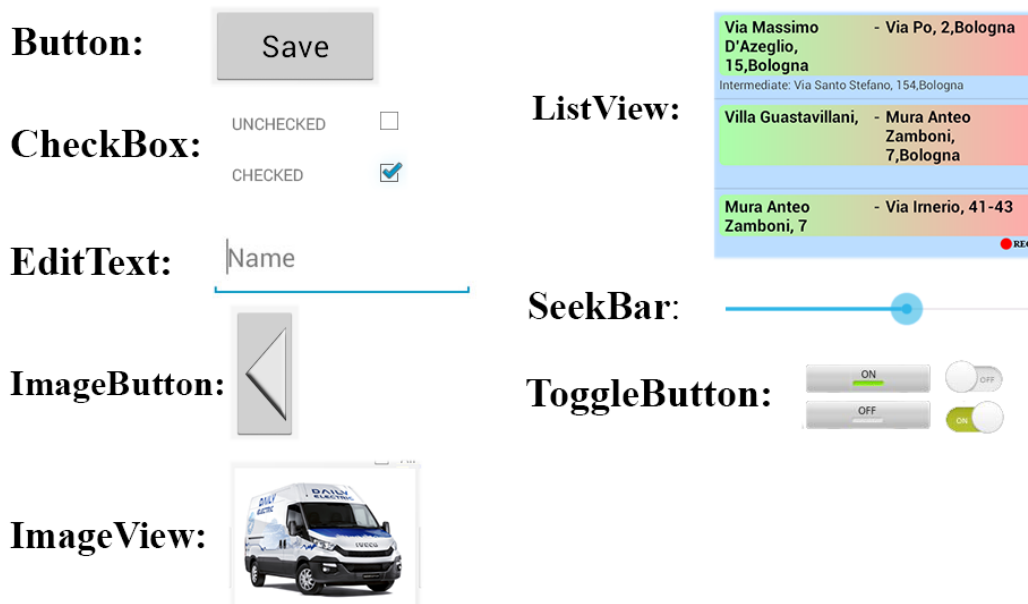


Figura 3.3: Widget utilizzati per WIA

- **ImageButton:** mostra un *Button* con un'immagine (anzichè del testo) che può essere premuto o cliccato dall'utente. (Nota: non è un'estensione di *Button*, ma di *ImageView*)

- **ImageView**: permette di visualizzare un'immagine all'interno di una *View*.
- **ListView**: permette la visualizzazione di informazioni attraverso una lista.
- **SeekBar**: una *SeekBar* è un'estensione della *ProgressBar* a cui è stata aggiunta un elemento trascinabile. L'utente può toccare l'elemento e trascinarlo a destra o sinistra per aumentare o diminuire un certo valore.
- **TextView**: mostra del testo all'utente modificabile con il suo metodo `setText()`.
- **ToggleButton**: pulsante con il significato di on/off.

Capitolo 4

WhatIfApp - (WIA)

L'idea che ha portato alla nascita di WhatIfApp, è stata quella di voler dare uno strumento per dare, finalmente, un'idea chiara di quella che è la sostenibilità attuale di un EV.

In particolare l'utente, utilizzando l'applicazione, sarà in grado di creare, manualmente o registrandole, le tratte da lui percorse quotidianamente, e di simulare quella tratta con un EV per valutarne l'idoneità rispetto ad un veicolo a benzina/diesel, facendo, possibilmente, un paragone con altri EV.

Di seguito verranno mostrate le funzionalità messe a disposizione dall'applicazione.

4.1 Tipologie di tratte

In primis va spiegato su quali tipologie di tratte (chiamate Trip) vengono fatte le simulazioni:

1. **Offline TRIP:** percorso semplice composto da Partenza, Destinazione e una serie di Intermezzi (utili per rendere più preciso il percorso da simulare nel caso in cui quello calcolato dall'applicazione non fosse sufficientemente preciso rispetto al percorso

reale o semplicemente per segnalare delle deviazioni di percorso rispetto a quello standard).

2. **Recorded TRIP**: percorso semplice registrato in tempo reale dall'utente durante il viaggio.
3. **Journey**: collezione di trip, utile per la simulazione di più tratte nel corso di una o più giornate.

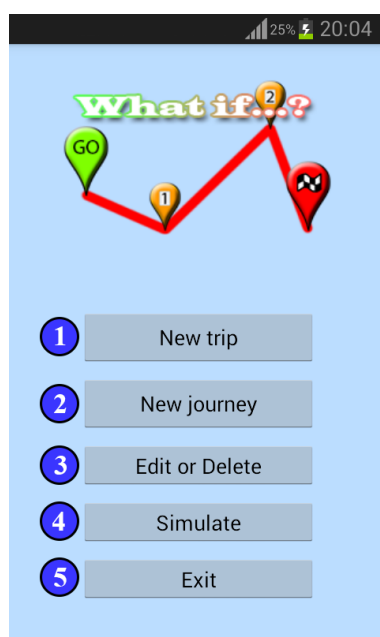


Figura 4.1: Menu principale di WhatIfApp

4.2 Creazione di un Offline TRIP

Dal menù principale, selezionando il Button(fig.4.1[1]) "New trip", ci verrà data la possibilità, tramite l'apertura di un Dialog, di creare un Offline TRIP o un Recorded TRIP (fig.4.2).

Selezionando il Button "Offline TRIP" verrà aperta l'activity relativa.

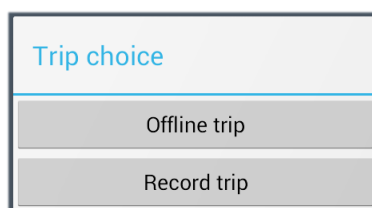


Figura 4.2: Dialog per la selezione del tipo di TRIP da creare.

4.2.1 Inserimento delle locazioni (Partenza, Destinazione, Intermezzi)

Ci sono due modalità differenti per l'inserimento della Partenza e la Destinazione:

1) Inserimento tramite Digitazione manuale

Selezionando i relativi EditText(fig.4.3[1]), verrà mostrata la tastiera digitale per la digitazione delle informazioni utili alla ricerca della locazione.

Una volta digitata la Partenza o la Destinazione, a sinistra del rispettivo EditText, verrà mostrato un Button(2) per la ricerca della locazione inserita con relativo piazzamento di un Marker nella posizione trovata. L'utente, in questo modo, potrà verificare se la posizione trovata coincide con quella desiderata, in caso contrario sarà comunque possibile affinare la ricerca inserendo più informazioni.

La locazione viene cercata tramite la classe Geocoder, utilizzando la funzione:

```
getFromLocationName (String locationName, int maxResults)
```

dove *locationName* è l'informazione inserita nell'EditText dall'utente e *maxResults* il massimo numero di risultati che dovranno essere restituiti.

Il risultato ottenuto sarà una lista di Address contenenti le coordinate

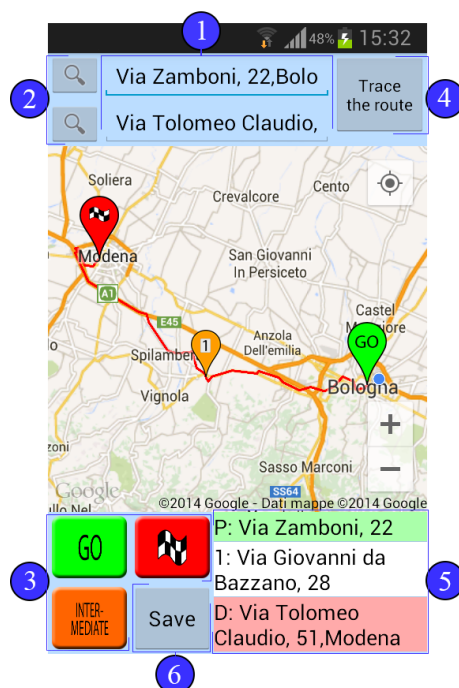


Figura 4.3: Activity per la creazione del'Offline TRIP

geografiche (latitudine, longitudine) della locazione.

Verrà preso in considerazione il primo Address presente in lista.

2) Inserimento tramite ToggleButton

Un metodo alternativo per l'inserimento della Partenza e/o la Destinazione e l'unico metodo per l'inserimento degli intermezzi è tramite i ToggleButton(fig.4.3 [3]) e l'interazione con la mappa.

Premendo su uno dei tre ToggleButton sarà possibile, tramite successivo tocco su mappa, l'inserimento diretto del marker nella posizione selezionata.

Il tipo di marker posizionato sarà diverso in base a quale dei tre ToggleButton risulterà attivo durante il tocco sulla mappa.

La locazione viene cercata, come per la digitazione manuale, tramite la classe geocoder utilizzando però questa volta la funzione:

```
getFromLocation (double latitude, double longitude, int maxResults)
```

dove *latitude* è la latitudine della locazione toccata su mappa, *longitude* è la longitudine e *maxResults* è il massimo numero di risultati che dovranno essere restituiti.

Al contrario della *getFromLocationName*, che cercava le coordinate geografiche della locazione in base ad un "nome" inserito nelle Edit-Text, la *getFromLocation*, a partire dalle coordinate geografiche, restituisce una lista con via, numero civico, città, stato e provincia della location cercata.

Nota: Solo un ToggleButton alla volta potrà rimanere sullo stato ON, premere su un ToggleButton mentre è attivo un altro differente, disabiliterà quello attivato precedentemente.

Nota: Una volta inserito il marker tramite tocco sulla mappa, il ToggleButton attivo verrà messo su OFF.

4.2.2 Tracciare il percorso

Se sia la Partenza che la Destinazione sono stati inseriti, sarà possibile, premendo sul Button((fig.4.3[4]) "Trace the route", tracciare il percorso(in rosso) tra le varie locazioni trovate.

La ricerca del percorso verrà effettuata tramite una query alle Google Directions API all'indirizzo:

```
https://maps.googleapis.com/maps/api/directions/json?...  
...origin=Origin&destination=Destination&key=API_KEY
```

che permettono il calcolo del percorso tra due o più punti con diverse modalità (in auto, a piedi, in bicicletta...).

L'applicazione cerca i percorsi calcolati esclusivamente per gli autoveicoli.

La risposta è in formato Json e contiene molte informazioni riguardo al percorso calcolato, quelle utili al progetto sono state le singole coordinate geografiche di ogni punto della tratta calcolata (per tracciare il percorso sulla mappa) e la durata (utile per la creazione dei Journey [vedi **cap.4-4**]).

4.2.3 Modifiche alla tratta

Sia prima che dopo la ricerca del percorso, è possibile spostare le locazioni già inserite o eliminarle.

Spostamento della locazione

È possibile modificare il luogo di una determinata locazione semplicemente spostando i relativi Marker presenti sulla mappa.

Nota: lo spostamento avverrà dopo un onLongClick(ovvero un tocco prolungato) sul marker.

Nota: Lo spostamento della Partenza o Destinazione è possibile anche cercando un'altra Partenza o Destinazione con i metodi citati sopra.

Eliminazione della locazione

Per l'eliminazione della locazione basterà un onLockClick sul rispettivo elemento nella ListView(fig.4.3[5]) dedicata al riepilogo del viaggio.

Nota: un Dialog ci chiederà la conferma dell'eliminazione..

4.2.4 Salvataggio della tratta

Una volta tracciato il percorso verrà abilitato il Button(6) per il salvataggio dell'Offline TRIP.

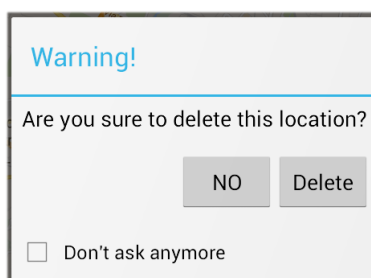


Figura 4.4: Dialog per la conferma dell'eliminazione della locazione

Premuto il Button, il TRIP verrà memorizzato in un DB nella forma:

- **_id**: Intero unico per ogni tratta memorizzata
- **Origin**: Stringa nella forma:

"Location@latitude,longitude"

Dove *Location* è il nome della locazione (Es. Via Zamboni, 22, Bologna, Italy).

- **Destination**: Stringa nella forma:

"Location@latitude, longitude".

- **Intermediate**: Stringa nella forma:

"Location@latitude,longitude#Location@latitude,longitude...".

- **Duration**: Intero che rappresenta la durata in secondi per effettuare il percorso.

4.3 Creazione di un Recorded TRIP

Dal menù principale, premendo il su "New TRIP", ci verrà data la possibilità di creare un Offline TRIP o un Recorded TRIP (fig.4.2).

Premendo su "Recorded TRIP" verrà avviata l'activity relativa alla registrazione del TRIP.

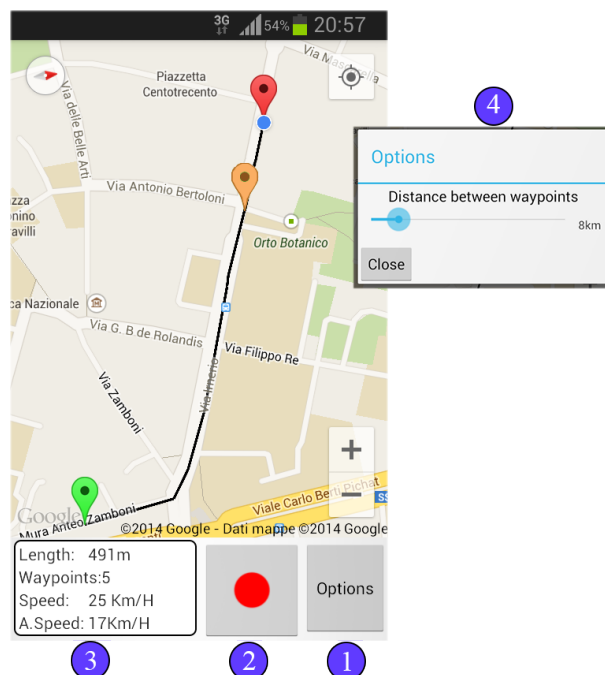


Figura 4.5: Activity per la creazione del Recorded TRIP

4.3.1 Registrazione della tratta

La registrazione del TRIP può avvenire solo a geolocalizzazione avvenuta.

Selezionando il Button(fig.4.5[1]) "Options" verrà mostrato un Dialog (fig.4.5[4]) da cui sarà possibile selezionare, tramite una SeekBar, il livello di precisione della registrazione, ovvero la distanza tra un waypoint e l'altro. Più sarà alta la precisione, maggiore saranno i tempi di simulazione. Se non viene impostata la precisione, allora verrà preso in considerazione un livello di precisione standard (10km).

Selezionare l'ImageButton(fig.4.5[2]) con il cerchietto rosso, avvierà la

registrazione del TRIP. Nel riquadro in basso a sinistra(fig.4.5[3]) vengono mostrati alcuni dati tra quelli registrati, ovvero la lunghezza del percorso registrato, il numero dei waypoints memorizzati, la velocità istantanea e la velocità media.

A registrazione avviata, vengono mostrati due ImageButton (fig.4.6), uno per mettere in pausa la registrazione e uno per terminarla, rispettivamente al posto dell'ImageButton per l'avvio e il Button "Options".



Figura 4.6: ImageButton per la pausa e lo stop della registrazione

Premere sull'ImageButton(fig.4.6[1]) per la pausa, verrà piazzato un marker per evidenziare la sosta e, inoltre, verrà chiesto all'utente di specificare il motivo della pausa (fig.4.7)(non obbligatorio). Una vol-

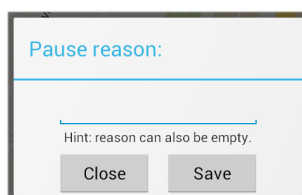


Figura 4.7: Dialog per specificare il motivo della pausa

ta messa in pausa la registrazione, l'utente potrà scegliere di riavviare la registrazione premendo sull'ImageButton con il cerchio rosso, che sarà, nel frattempo, apparso al posto di quello per la pausa. Sia che la registrazione sia avviata o che sia messa in pausa, l'utente potrà, in qualsiasi momento, decidere di interromperla selezionando l'Image-

Button (fig.4.6[2]). Subito dopo lo stop della registrazione, avverrà il salvataggio su DB di tutto ciò che è stato registrato

4.3.2 Salvataggio della tratta registrata

Una volta premuto l'ImageButton (fig.4.6 [2]), se il viaggio registrato ha una lunghezza di minimo 300 metri, verrà memorizzato in un DB nella forma:

- **_id**: Intero unico per ogni tratta memorizzata
- **Origin**: Stringa nella forma:

"Location@latitude,longitude"

dove *Location* è il nome della locazione.

- **Destination**: Stringa nella forma:

"Location@latitude, longitude"

- **Intermediate**: Stringa nella forma:

"Location@latitude,longitude#pause@latitude,longitude..."

- **Duration**: Intero che indica la durata del TRIP registrato in secondi.
- **Hour**: Double per la memorizzazione dei tempi di salvataggio di Partenza/Destinazione/Intermezzo.

4.4 Creazione di un Journey

Dal menù principale, premendo il Button "New Journey" verrà aperta l'activity relativa alla creazione del viaggio.

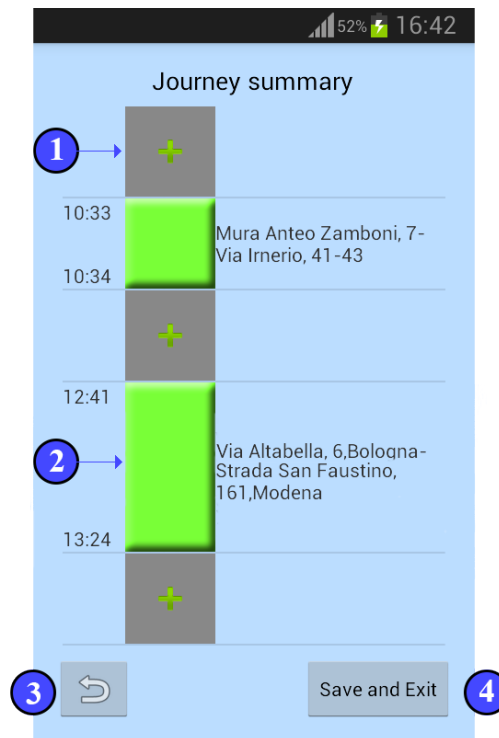


Figura 4.8: Activity per la creazione del Journey

4.4.1 Linea temporale

Per la rappresentazione del viaggio è stato scelto l'uso di una linea temporale, in questa maniera dovrebbe risultare più semplice per l'utente visualizzare e percepire in quale momento di una determinata giornata è stato collocata un determinato Trip, inoltre risulterà molto utile successivamente quando si andrà ad effettuare la simulazione, per il paragone tra il Journey prima e dopo essere stato simulato (EV Journey). La linea temporale è composta da riquadri(fig.4.8 [1]) che rappresentano le pause, ovvero i momenti in cui l'auto non viene utilizzata, e da riquadri(fig.4.8 [2]) che rappresentano i diversi trip. Entrambi i riquadri sono dei Button di lunghezza variabile in base alla loro durata.

4.4.2 Gestione del Journey

Cliccando sui riquadri che rappresentano le pause, verrà aperta un activity che permetterà all'utente di selezionare quale Trip (Offline o Registrato) già creato inserire nel Journey e in quale lasso di tempo.

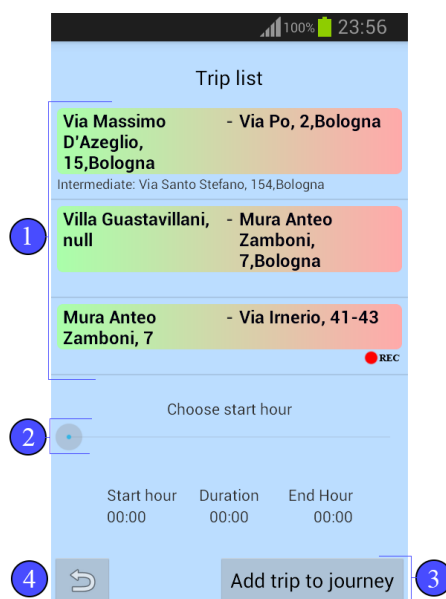


Figura 4.9: Activity per la selezione del trip da aggiungere al Journey

Selezionando uno tra i trip visualizzati nella ListView(fig.4.9 [1]), sulla TextView dedicata alla durata, verrà mostrata la durata in formato hh:mm. Inoltre verrà attivata la SeekBar(fig.4.9 [2]) per poter selezionare l'orario di partenza.

L'orario di arrivo viene impostato automaticamente.

Premendo il Button(fig.4.9 [3]) "Add trip to journey", il trip selezionato verrà aggiunto ad un DB temporaneo, utilizzato nella schermata principale per la gestione del journey, per mostrare il journey sulla linea temporale.

Premendo l'ImageButton(fig.4.9 [4]), l'Activity verrà chiusa senza salvare nulla.

Una volta terminato il journey, premendo il Button(fig.4.8 [4]), verrà salvato in un DB nella forma:

- **_id**: Intero unico per ogni Journey.

- **Start**: Stringa nella forma:

"StartLoc@lat,lng@Intermediate@lat,lng ... @EndLoc@lat,lng"

- **End**: string nella forma:

"StartLoc@lat,lng@Intermediate@lat,lng ... @EndLoc@lat,lng"

- **IntermediateTrip**: String nella forma:

"StartLoc@lat,lng@Inter@lat,lng ... @EndLoc@lat,lng#StartLoc
..."

- **Hour**: Lista di interi contenenti l'orario di inizio e fine di ogni trip nel journey.

Nota: l'aggiunta di un trip in una pausa compresa tra altri due trip, porrà dei limiti al lasso di tempo in cui quel trip potrà essere inserito (Es: tra un trip che finisce alle 18 e uno che inizia alle 24, il nuovo potrà essere collocato tra le [18:00] e le [24:00 meno la durata del viaggio], se le 24:00 meno la durata del viaggio sarà minore delle 18, allora la tratta non potrà essere inserito e quindi non apparirà tra i trip disponibili .

Nota: Se un trip ha una partenza eccessivamente distante dall'arrivo del precedente trip, o se il suo arrivo è eccessivamente distante dalla partenza del successivo trip, il trip preso in considerazione verrà considerato come non valido e pertanto non verrà mostrato.

4.5 Modifica/Elimina di un trip/journey

Dal menù principale, selezionando il Button "Update/Delete trip/journey" verrà aperta l'activity dedita alla modifica e/o alla eliminazione dei vari trip o journey creati sin ora.

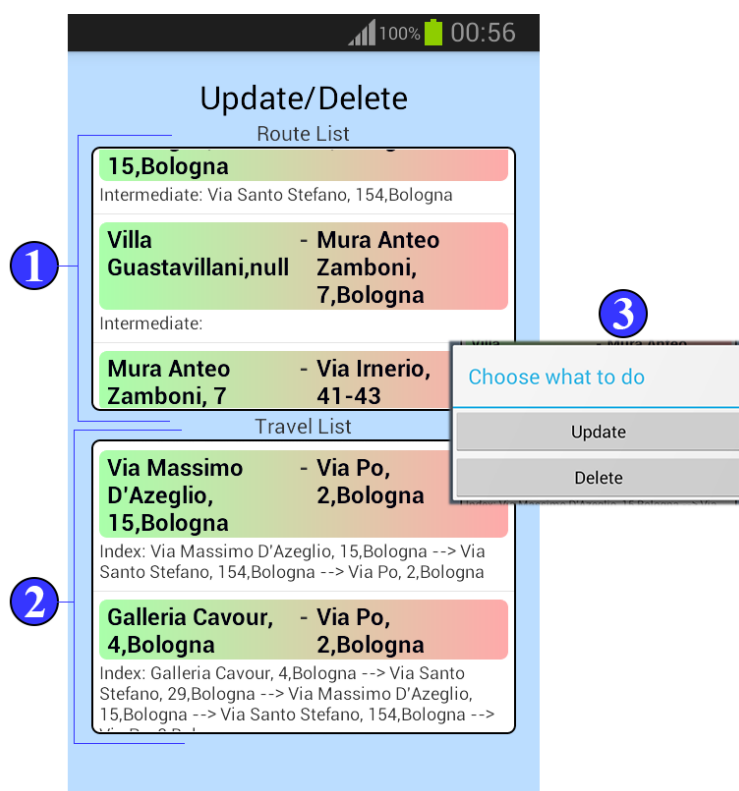


Figura 4.10: Activity per la modifica/eliminazione di un trip/journey

Nella ListView(fig.4.10[1]) in alto verranno visualizzati gli Offline TRIP e i Recorded TRIP, mentre nella ListView(fig.4.10[2]) in basso verranno mostrati i Journey.

Selezionando un trip o un Journey, verrà aperto un Dialog(fig.4.10[3]) per selezionare la modifica o l'eliminazione dell'elemento selezionato. Non sarà possibile modificare un Trip registrato.

Selezionando il Button "Update" verrà aperta l'activity per la gestione degli Offline TRIP nel caso in cui l'elemento selezionato era un trip, o l'Activity per la gestione dei Journey altrimenti.

Le Activity verranno aperte con i viaggi già impostati.

Salvando i trip/journey per la modifica, non saranno aggiunti altri elementi nei rispettivi DB, ma verranno modificati quelli già esistenti.

Premendo, invece, il Button "Delete", verrà semplicemente rimosso l'elemento dal DB e l'Activity verrà refreshata per permettere la corretta visualizzazione degli elementi ancora esistenti.

4.6 Simulazione di un Trip/Journey

Dal menù principale, premendo il su "Simulate Trip/Journey" verrà aperta l'activity relativa alla simulazione dei vari Trip o Journey precedentemente creati.

I trip memorizzati verranno visualizzati nella ListView (fig.4.11[1]) in alto, mentre i Journey nella ListView (fig.4.11[2]) in basso.

Selezionato il trip o il journey da simulare e selezionando il Button (fig.4.11[3]) "Next", verrà aperta un Dialog per l'inserimento dei parametri di simulazione, tra i quali:

- **Auto per la simulazione (EV):** tramite i due ImageButton (fig.4.12[2]) "Next" e "Previous" è possibile selezionare l'EV per la simulazione.
Gli EV attualmente selezionabili sono: Daily Electric,500e,Nissan Leaf,i-MiEV,Tesla S,Smart ED,Renault Fluence ZE,BMW Active E.
È anche possibile scegliere di effettuare la simulazione con tutti gli EV disponibili spuntando la CheckBox (fig.4.12[1]) "All".
- **Carica iniziale:** Tramite la SeekBar (fig.4.12[3]) è possibile impostare la carica iniziale assegnata all'auto per la simulazione.



Figura 4.11: Activity per la selezione del Trip/Journey da simulare

- **Minimo della carica al termine della simulazione:** Tramite la SeekBar (fig.4.12[4]) è possibile indicare il minimo di carica che l'EV dovrà avere al termine della simulazione.

Premendo sul Button (fig.4.12[5]) "Simulate" verrà avviata la simulazione.

Quando la simulazione sarà terminata in base all'elemento scelto (Trip o Journey) verrà aperta un Activity diversa.

4.6.1 Simulazione di un Trip

Per la simulazione di un trip vengono eseguite N-1 query, dove per N si intende il numero di waypoints che compongono il trip. La seconda

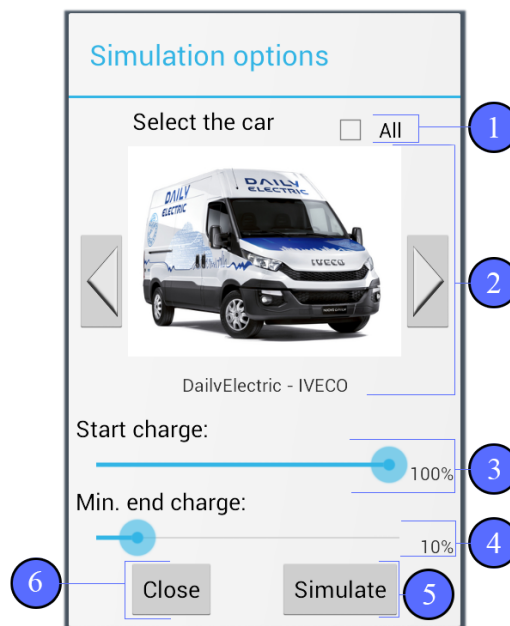


Figura 4.12: Dialog per la scelta dei parametri di simulazione

query verrà chiamata al termine della prima e così via.

Per richiedere la simulazione al servizio web di Route Planning viene fatta una richiesta GET nella seguente forma:

```
http://hostname-macchina.cs.unibo.it:8080/route/  
:origin/:destination/:vehicle/:SOC/:threshold/:thresholdArrival/
```

i parametri seguono i : e sono:

- **origin**: Stringa nella forma "via, città, ..." o in coordinate geografiche.
- **destination**: Stringa nella forma "via, città, ..." o in coordinate geografiche.
- **vehicle**: Stringa contenente il nome del veicolo
- **SOC**: Double indicante lo stato di carica iniziale.

- **threshold**: Double che indica la carica minima della batteria dell'EV per arrivare alla prima colonnina di ricarica utile.
- **thresholdArrival**: Double che indica la carica minima della batteria dell'EV per arrivare a fine viaggio.

Se l'elemento selezionato per la simulazione è un trip, verrà direttamente aperta l'Activity dedicata al riepilogo della simulazione.



Figura 4.13: Activity per la visualizzazione dei dati raccolti durante la simulazione

Nella GoogleMap(fig.4.13[1]) in alto apparirà la tratta simulata (in rosso), delimitata da un marker per la partenza, uno per la Destinazione ed, eventualmente, un marker per indicare il luogo dove è avvenuta la ricarica dell'auto.

In basso(fig.4.13[2]), invece, verranno visualizzati i dati raccolti durante la simulazione, ovvero:

- **Carica iniziale:** La carica iniziale della batteria dell'EV per la simulazione (in KWh e in %).
- **Consumo:** Il valore in KWh e in percentuale del consumo che l'EV ha dovuto sostenere durante la simulazione(in KWh e in %).
- **Carica finale:** La carica con cui l'EV è arrivato a fine simulazione (in KWh e in %).
- **Durata:** Indica il tempo impiegato dall'EV per terminare il percorso.
- **Distanza:** Indica la distanza in km percorsa dall'EV.
- **Charge:** Indica la quantità di energia consumata durante il tragitto (in KWh e in %).
- **N. of charge:** Indica il numero di volte in cui l'EV ha dovuto fare una sosta per ricaricarsi.

Premendo sugli ImageButton(fig.4.13[3]) verranno visualizzati in una ListView(fig.4.13[4]), al posto dei dati appena elencati, informazioni riguardanti il percorso che è stato appena simulato.

4.6.2 Simulazione di un Journey

Se l'elemento selezionato per la simulazione è un Journey, verrà aperta un Activity che mostrerà all'utente due linee temporali in due ListView differenti.

La prima a sinistra(fig.4.14[1]) rappresenta il Journey pianificato precedentemente; mentre quella a destra(fig.4.14[2]) rappresenta il Journey

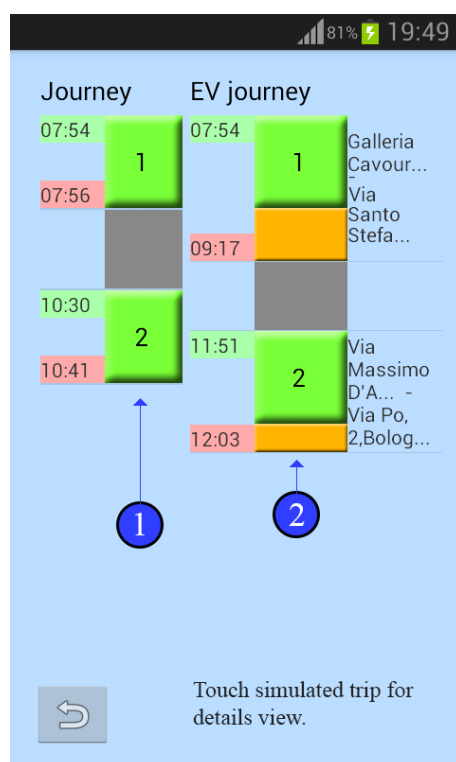


Figura 4.14: Activity per il paragone tra i due Journey

dopo la simulazione con un EV (EV Journey).

Le differenze di durata dei diversi trip che compongono i journey vengono visualizzati dai riquadri color arancio.

Le differenze di tempo possono variare, ad esempio, se l'EV ha dovuto sostare per una ricarica.

Le pause verranno prese in considerazione per eventuali ricariche, limitando, quindi, il ritardo accumulato per una ricarica.

Cliccando sui riquadri, nell'EV Journey, dei trip, verrà aperta l'Activity, come per le simulazioni dei Trip, per far visualizzare, nel dettaglio, quel particolare trip del Journey.

La simulazione di un Journey ha una caratteristica in più rispetto a quella di un semplice Trip. Durante una pausa, se quest'ultima viene fatta nelle vicinanze (max 1km di distanza) di una colonnina di ricarica, la

simulazione sfrutterà questa pausa per ricaricare l'auto il più possibile, evitando, in questo modo, di doversi ricaricare nel successivo Trip.

Conclusioni

A progetto concluso, WhatIfApp si propone come un valido e utile strumento per la valutazione della mobilità su determinati percorsi utilizzando gli EV.

Sin da subito, infatti, l'utente potrà rendersi conto di quali siano gli effettivi consumi e le tempistiche di ricarica, se i tratti di strada percorsi ogni giorno, per andare a lavoro o per qualsiasi altro motivo, siano stati effettuati con un EV, anzichè con un veicolo a benzina.

Le migliorie apportabili al servizio sono numerose, alcune di esse sicuramente potrebbero essere:

- Inserimento di un numero maggiore di EV disponibili per la simulazione.
- Implementare un sistema per riconoscere automaticamente le pause, nella registrazione del Trip, in modo tale che l'utente non debba ricordarsi necessariamente ogni volta di mettere in pausa o in stop la registrazione.
- Rendere disponibili maggiori opzioni utili per la simulazione dei vari Trip o Journey, come ad esempio la possibilità di calcolare percorsi alternativi a quelli pianificati, ottimizzati in base al consumo.
- Importazione dell'applicazione su altri SO.

Elenco delle figure

2.1 Architettura di Internet of Energy	2
2.2 Architettura del web service per il route planning	4
2.3 Icona di iEV	5
3.1 Logo Android	7
3.2 Il ciclo di vita di un'activity in Android	12
3.3 Widget utilizzati per WIA	15
4.1 Menu principale di WhatIfApp	18
4.2 Dialog per la selezione del tipo di rotta da creare	19
4.3 legenda elenco figure	20
4.4 Dialog per la conferma dell'eliminazione della locazione	23
4.5 Activity per la creazione del Recorded TRIP	24
4.6 ImageButton per la pausa e lo stop della registrazione	25
4.7 Dialog per specificare il motivo della pausa	25
4.8 Activity per la creazione del Journey	27
4.9 Activity per la selezione del trip da aggiungere al Journey	28
4.10 Activity per la modifica/eliminazione di un trip/journey	30
4.11 Activity per la selezione del Trip/Journey da simulare	32
4.12 Dialog per la scelta dei parametri di simulazione	33
4.13 Activity per la visualizzazione dei dati raccolti durante la simulazione	34
4.14 Activity per il paragone tra i due Journey	36

Elenco delle tabelle

3.1	Elenco delle principali versioni di Android	9
-----	---	---

Bibliografia

- [1] <https://developers.google.com/maps/documentation/android/>
- [2] <https://developers.google.com/maps/documentation/directions/>
- [3] Internet of Energy (IoE) - ARTEMIS European Project.
Project Website: <http://www.artemis-ioe.eu>
- [4] *L. Bedogni, L. Bononi, A. D'Elia, M. Di Felice, M. Di Nicola, T. Salmon Cinotti Driving Without Anxiety: a Route Planner Service with Range Prediction for the Electric Vehicles, to appear on Proceedings of the 3rd International Conference on Connected Vehicles and Expo (IEEE ICCVE 2014), 3-7 November 2014, Vienna, Austria.*
- [5] *L. Bedogni, L. Bononi, A. D'Elia, M. Di Felice, S. Rondelli, T. Salmon Cinotti A Mobile Application to Assist Electric Vehicles' Drivers with Charging Services, to appear on Proceedings of the 8th International Conference and Exhibition on Next Generation Mobile Applications, Services, and Technologies (IEEE NGMAST 2014), 10-12 September 2014, Oxford, UK.*
- [6] <http://it.wikipedia.org/wiki/Android>
- [7] Info vendite smartphone: <http://www.ansa.it>
- [8] Dalvik Virtual Machine: <http://it.wikipedia.org/wiki/Android> Sezione DalvikVirtualMachine

- [9] Geocoder:
<http://developer.android.com/reference/android/location/Geocoder.html>
- [10] Android Widget: *Android 3 Guida per lo sviluppatore - Massimo Carli - APOGEO*
- [11] Android Widget:
<http://developer.android.com/reference/android/widget>

Ringraziamenti

Vorrei innanzitutto ringraziare i miei genitori, poichè, senza i loro sacrifici, oggi sicuramente non sarei qui.

Ringrazio Antonia Lopreside, per l'infinito sostegno e supporto che in 8 anni è riuscita a darmi e senza il quale oggi mi sentirei perso.

Ringrazio il mio relatore, Luciano Bononi, e i miei correlatori, Luca Bedogni e Marco Di Felice, per la loro infinita pazienza e disponibilità, e per avermi dato l'opportunità di lavorare a questo bellissimo progetto.

Ringrazio i miei fratelli, per avermi concesso sempre qualcuno con cui litigare.

Ringrazio i miei amici e colleghi, Antonio Carbonara, Emanuele Visconti, Lara Vigneux e Danilo Pecoraro, per l'aiuto e per aver reso divertente lo studio di ogni singolo esame sostenuto.

Ringrazio i miei amici che, oggi, sono venuti sin qui dalla Puglia, Arturo Mavellia, Alessandro Antonelli e Katia Losapio, non lo dimenticherò mai.

Ringrazio i miei amici presenti qui oggi: Davide, Rita e Giulia e ringrazio i miei amici, tutti, sia quelli di vecchia data che i più recenti che non sono potuti essere presenti, grazie per ogni singolo momento passato insieme a voi.

Ringrazio i miei attuali ed ex coinquilini, David, decisamente il più pazzo, Emanuele, Tancredi, Federico, Salvatore e Piero, per aver reso indimenticabile la mia permanenza a Bologna sino ad oggi.

Ringrazio i miei parenti e i loro "Ma quando ti laurei?", per l'avermi

messo fretta ad ogni festività.

Ringrazio i miei nonni, in particolare, nonno "Ciccio", a cui purtroppo non potrò mai più far sapere che anch'io ce l'ho fatta.

Ringrazio i miei cugini, che spesso e volentieri sono stati più 'amici' che 'parenti'.

Ringrazio Sabina Di Stato e Francesco Lopreside anche voi giunti sin qui dalla Puglia, per avermi sempre fatto sentire come in una seconda famiglia.

Infine ringrazio i miei amici di nerding: NeriDK, Babinx, Sagrinx, DÃ,y, Recklash, PrometheusDK, Suppaman, Vontrevor e StarFlare, grazie per aver ritardato il più possibile la mia laurea. Mille grazie a tutti voi che, giorno dopo giorno, avete fatto di me, quello che sono oggi.