

Università degli Studi di Napoli Federico II
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica LM

Testina Applicazione Android

RedMoon

Candidato:
Sajmir Rusi

Relatore
Prof. Porfirio Tramontana

Introduzione

L'idea:

Oggetto dell'elaborato è lo sviluppo di un applicazione Android che offre all'utente diversi dati, come l'orario di sorgere e tramonto del sole, le previsioni meteo nella città nella quale si trova il dispositivo, il verificarsi dell'evento Luna Rossa e altri dati supplementari di interesse astronomico.

L'applicazione è nata con l'idea principale di segnalare all'utente il giorno in cui si può vedere la luna rossa dalla propria città, dopo di che si è pensato di estendere tale funzionalità.

Attualmente i dati meteorologici vengono recuperate dalle API Yahoo, tramite le quali in base alla città viene fatta una richiesta specifica e ricevute le informazioni meteo.

Sviluppo:

Per sviluppare l'applicazione è stato pensato di suddividere il lavoro in fasi differenti:

- Studio delle API Yahoo
- Analisi della fonte di dati dal sito <http://www.marcomenichelli.it/>
- Studio dell'architettura Android e delle principali tecniche di sviluppo
- Studio di fattibilità; progettazione delle interfacce e dei casi d'uso
- Implementazione e testing

L'applicazione

Per lo sviluppo dell'applicazione Android sono stati utilizzati i seguenti strumenti:

- Android SDK, kit di sviluppo che fornisce le librerie API e gli strumenti necessari a costruire, testare e fare il debug di una app Android:
 - Eclipse + Android Developer Tool (plugin ADT) una piattaforma questa per la creazione di applicazioni Android
 - Strumenti di supporto di Android
 - Emulatore Android
- Yahoo API <http://weather.yahooapis.com/forecastrss>
- Fonte dati astronomici

Panoramica:

L'applicazione è strutturata in tre sezioni differenti ognuna delle quali effettua una o più richieste alle API utilizzate o alla fonte dati. Per ottimizzare l'utilizzo delle risorse si è pensato di sfruttare un unico Client http per le diverse chiamate che vengono inviate dalla Applicazione.

Le interfacce di interazione con l'utente vengono gestite da una Activity principale il cui ciclo di vita è stato modellato in modo tale da avere una fruibilità della Applicazione.



Funzionamento:

Il funzionamento dell'Applicazione è stato pensato in due modalità differenti, le quali possono essere scelte nella menu impostazioni della App stessa.

- Automatico, si utilizza la geolocalizzazione tramite il gps del dispositivo. Poiché il margine di errore della località per l'App non è importante, basta recuperare informazioni riguarda la città dove il dispositivo si trova.
- Manuale, in questo caso tramite il menu impostazioni si può selezionare la regione desiderata dopo di che compare la lista con le città, si seleziona quella per la quale si vogliano avere le informazioni meteorologiche e astrologiche.



Una volta impostato la modalità di funzionamento, si possono ricevere i dati di interesse facendo pressione sul button "Ricevi Dati".

Yahoo API

Le Yahoo Application Programming Interface consentono di ricevere informazioni meteorologiche aggiornate in base alla locazione. Le informazioni vengono scaricate con una semplice Http Get nella forma:

<http://weather.yahooapis.com/forecastrss?w=2502265&u=c>

dove il parametro w è un identificativo univoco del luogo, chiamato anche "Where On Earth Identifiers" tale parametro l'applicazione lo aggiunge in modo dinamico alla url di base.

Invece il secondo parametro è opzionale, consente di specificare l'unità di misura delle temperatura, u=c imposta l'unità in °C nel caso in cui non viene specificato l'unità e quella di default in Fahrenheit.

Come risposta alla richiesta Http Get fatta in questo modo viene ricevuto un file xml con delle informazioni aggiornate sul meteo.

Per presentare in modo user friendly le informazioni ricevute è stato utilizzato un SAX parser, andando a estrapolare dal xml solo quelle informazioni di interesse per l'applicazione.

Come primo approccio per la ricezione dei dati astrologici è stato pensato di usare il sito <http://www.marcomenichelli.it/>. Tramite chiamate Http Post all'interno delle quale si poteva specificare l'Anno, il mese, la nazione e la città si aveva in risposta una pagina html molto mal organizzata ed è stato impossibile eseguire tale approccio per la ricezione delle informazioni e il successivo parsing.

Struttura del codice

Application

Per ottimizzare le risorse occupate durante il ciclo di vita delle activity è stato pensato di creare un unico HttpClient condiviso tra le diverse activity. Sfruttando il fatto che le diverse activity di una Applicazione anche se hanno cicli di vita differenti condividono un oggetto comune di tipo Application. Questo è il luogo ideale dove inserire dei oggetti che vengono condivisi dalle diverse activity della nostra App.

Inoltre nel contesto della Application viene fatto l'overloading di altri due metodi importanti: onLowMemory() e onTerminate(), la prima consente di chiudere la connessione Http nel caso in cui ci sia insufficienza di memoria nel dispositivo, la seconda invece si occupa di chiudere in modo corretto la connessione Http.

RedMoon

La classe RedMoon estende la classe TabActivity. La classe TabActivity è stata deprecata in API Level 13 ma ancora largamente utilizzata, per questo motivo non sono state utilizzate le Fragments, inoltre con l'utilizzo di Fragments bisognava una libreria di supporto per una reverse compatibility.

All'interno di questa classe viene istanziato anche il menu dell'applicazione, con tre diverse opzioni:

- Settings che consente di impostare la modalità di funzionamento(ancora in fase di sviluppo)
- About Us che mostra delle informazioni riguardo l'applicazione.

- Ed il menu Esci che consente di terminare l'applicazione

Terra, Sole, Luna

Sono tre classi che ereditano dalla classe Activity, al momento solo l'opzione Terra è stata implementata e consente di ricevere dei dati sul meteo.