*Fabris Alberto* *5CIA, 01/02/2020*

**Android App Development - EasyRSS**

**Specifiche Tecniche**

**Descrizione**

EasyRSS permette una visualizzazione rapida dei cosiddetti Feed RSS, presenti su molti siti. La scelta è ricaduta su questo argomento dato che mi risulta utile perché mi permette di seguire le fonti che consulto più spesso in modo più veloce.

**Dati Tecnici Generali**

Choose the features and components included in your app:

* Activity (specify the number of activities in you app)

2 Activity, una Main che utilizza Fragment, e una News che visualizza un RSSItem.

* Service (specify the operations performed by the services)  
  …………………………………………………………………………………………………………………………………
* Broadcast Receiver (specify the purpose)  
  …………………………………………………………………………………………………………………………………
* Content Provider (specify the type of contents)  
  …………………………………………………………………………………………………………………………………
* Intent (to pass data between activities, to run another app, to run a service…)

Intent per avviare la News Activity, per condividere un Link, oppure per aprire un app.

* Fragments (specify the purpose)

Fragment delle schermate Home e Settings.

* AsyncTask (specify the purpose)

AsynkTask per gestire la Response di OkHttp

* Threads and Handlers (specify where you use threads and what for)

Thread per Ottenere e Visualizzare la pagina RSS

* SQLite database engine (specify the kind of data stored into SQLite db)  
  …………………………………………………………………………………………………………………………………
* External DBMS connection (MySQL, PostgreSQL, or specify others..)  
  …………………………………………………………………………………………………………………………………
* Internet Connectivity (specify the purpose)

Utilizzata per ottenere il file RSS corrispondente al feed

* Geo-location (specify the purpose)  
  …………………………………………………………………………………………………………………………………
* Localization (specify languages)  
  …………………………………………………………………………………………………………………………………
* Multiple device layout and resolution support (tablet, smartphone, ..specify)

App Testata su Smartphone e Tablet per renderla graficamente corretta.

* Hardware features (WiFi, bluetooth, camera, GPS, sensors, or specify others)  
  …………………………………………………………………………………………………………………………………
* Google Mobile Services (specify the service and the purpose)  
  …………………………………………………………………………………………………………………………………
* Web application/web service interaction (specify whether your app exchanges data with a web application or a web service, and the purpose of that)  
  …………………………………………………………………………………………………………………………………
* Third party libraries (does you app need external software components? specify)

OkHTTP, Gson

* Other embedded or involved technologies (QR codes, RFID and NFC, 3D graphics, data encryption, multicast streaming, vpn, specify…)  
  …………………………………………………………………………………………………………………………………
* Other (specify)

TinyDB, Utilizza le SharedPreferences ma permette di salvare anche Oggetti

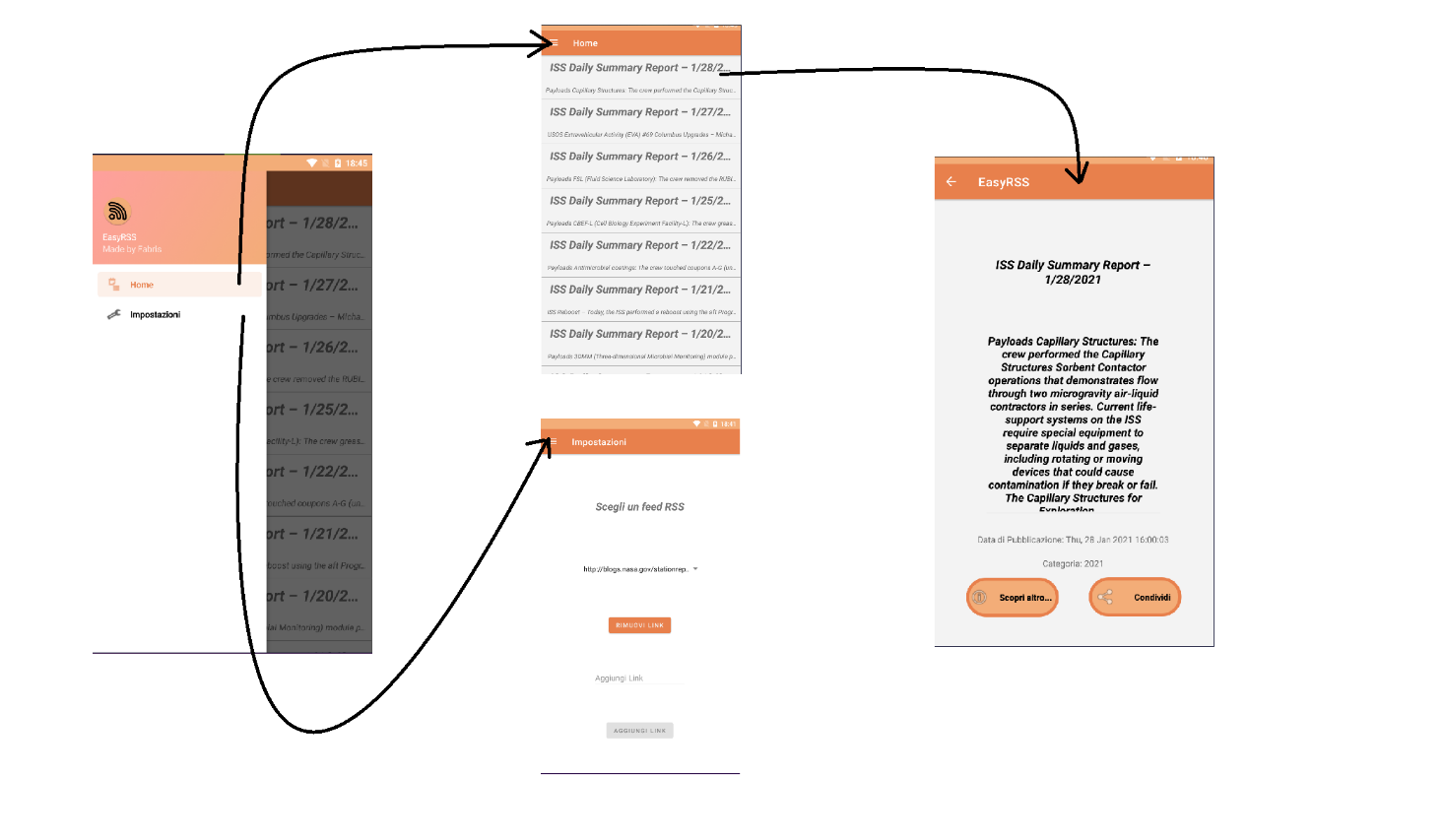
**Features**

EasyRSS permette di avere una lista di feed seguiti spesso, e di poter scegliere quale feed visualizzare. Inoltre all’avvio carica direttamente l’ultimo feed visualizzato, e tramite il click di un elemento della lista è possibile visualizzare i dettagli e condividerlo. Secondo me il vantaggio di quest’app è la velocità e la semplicità nell’utilizzo rispetto alle altre app concorrenti.

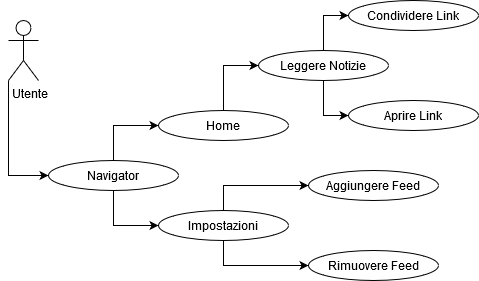
**App structure overview**

Explain the logical structure of you application. What are the building blocks? What are their functions? How do they interact with each other? You can enhance the explanation with a *wireframe* and/or a *use case diagram* (see examples below).

***Wireframe***



***Use case diagram***



**Code fragments**

Esempio di Richiesta HTTP

//Richiesta HTTP GET  
OkHttpClient ok = new OkHttpClient();  
Request request = new Request.Builder()  
 .url(url)  
 .build();  
  
//Gestore risposta  
ok.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Request request, IOException e) {}  
  
 @Override  
 public void onResponse(Response response) throws IOException {  
 if (response.isSuccessful()) {  
 String rss\_page = response.body().string();  
 }  
 }  
});

Esempio Parsing XML

Document doc = builder.parse(input);  
  
//Prendo una lista di tutti i nodi "item"  
NodeList nList = doc.getElementsByTagName("item");  
  
//Per ogni nodo si suppone che sia un RSSItem, e quindi aggiungo alla lista corrente il nuovo item  
for(int i = 0; i < nList.getLength(); i++){  
 this.add(new RSSItem(nList.item(i)));  
}

//Spezzone nella classe RSSItem

public void parseNode(Node no){  
 Element n = (Element)no;  
 try {  
 this.title = n.getElementsByTagName(*fieldnames*[0]).item(0).getTextContent();  
 } catch(NullPointerException npe){  
 this.title = "";  
 }  
 //[…] parti omesse, ripetitive  
 try {  
 this.pubDate = n.getElementsByTagName(*fieldnames*[4]).item(0).getTextContent();  
 }catch(NullPointerException npe){  
 this.pubDate = null;  
 }  
}

**Development**

Link alla repository GitHub: http://github.com/0fabris/tecno\_repo/tree/EasyRSS

Target API level: 29

Minimum API level: 16

IDE: Android Studio

Man-hours: 10h

***Problems and difficulties***

A mio avviso il problema principale è stato l’utilizzo degli AsyncTask per le chiamate HTTP.

***Reported Bugs***

Se può essere considerato un bug: il ricaricarsi del feed ogni volta che si gira lo schermo oppure quando cambia il fragment

***Further development***

Il progetto potrà sicuramente avere sviluppi futuri. Ad esempio una funzionalità da aggiungere è la gestione delle notizie preferite, salvandole sul dispositivo.

***Self-rating***

Penso che EasyRSS possa valere 4 stelle.

**References**

* Diapositive su Classroom
* StackOverflow (Vari Articoli)
* developers.google.com
* github.com