Homework 5, Distributed Applications Using Android

Network Programming, ID1212

Vasileios Charalampidis, vascha@kth.se

December 11, 2017

1 Introduction

The purpose of this fifth assignment is to design, develop and deploy an Android app using the Android SDK, in order to become familiar with it and also with the IDE development tools.

To meet these goals, an android application was deployed to handle the case of an RSS reader. In this application, the user has the option of typing any RSS url feed, in order to retrieve the last titles of this specific feed. After the successful fetching of the titles, if the user touches one of those, then he is redirected to the specific article through his selected browser.

The deployed program meets some specific requirements, so as to present an acceptable solution for this assignment. The first requirement is that some part of the application (e.g. clients) must run on Android devices or emulators of such devices. Some part (e.g. a server) may, but is not required to, execute on a computer connected to the Internet. There must be some communication from/to the nodes of the application running on Android devices. The second requirement is that the Android app must have a responsive user interface, which means it must be multithreaded. The user must be able to give commands, for example to quit the program, even if the client is waiting for a message from the server. The third requirement is that the Android app must have a GUI for user interaction developed using UI Views in Android. The fourth requirement is that the server, if there is a server, must be able to handle multiple clients playing concurrently, which means it must be multithreaded. The last requirement is that the user interface must be informative. This means that the current state of the program must be clear to the user, and the user must understand what to do next.

2 Literature Study

In order to accomplish this assignment, it was very important at the beginning to read carefully the application scenario and all its requirements. After the realization of the given context, the given material i.e. code and videos, at canvas for this course, was more than helpful for the java implementation. The videos gave a good first overview on how to deploy a simple Android application. Except from the provided course material, information about Android development found on the web was also part of dealing with the application.

What became clear from the materials used, was that Android applications are easy to develop and deploy, as the tools provided for that purpose are plenty. There is also a lot of documentation to refer so that to solve any issues or misunderstandings.

3 Method

The Android implementation was created with "Android Studio" by Google. Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. (ref. here).

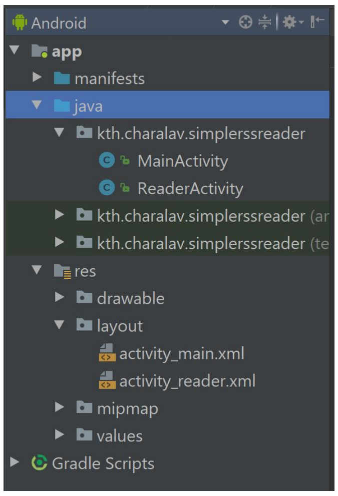
All the tests performed to the final deployed app were performed to a virtual Nexus 5X device, because of lack of physical android device.

To evaluate that the final solution met the requested requirements, several manual acceptance tests were performed to the android app e.g. fetch articles, handle errors, etc.

4 Result

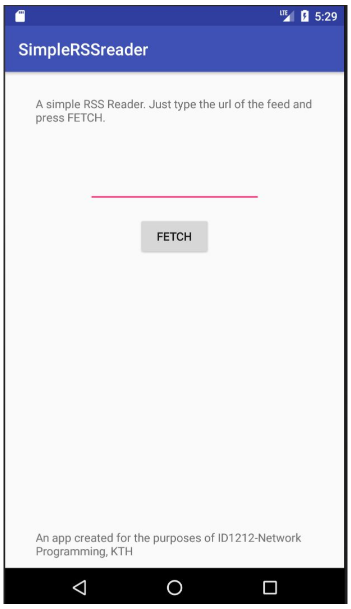
The architecture used for this assignment can be seen in Figure 1. It refers only to a client side. It is a well organized app with two activities that correspond to two different functionalities-client screens.

- The MainActivity class is the one that runs as soon as the client opens the app. Here the client is welcomed to the app and requested to type the desired rss feed to fetch (Figure 2). With the press of the FETCH button, the program checks if the client is actually connected to the internet (line 37). An appropriate permission was given in the Android Manifest file for accessing network state and internet (line 6-7). If the client has internet access, then a new intent is created (line 40), by including the input text of the client. With the creation of this intent, the ReaderActivity is called (line 42), while passing at the same time the input text of the client.



```
Android
▼ app
  ▶ manifests
  ▼ java
    ▼ kth.charalav.simplerssreader
        MainActivity
        ReaderActivity
    ▶ kth.charalav.simplerssreader (ar
    ▶ kth.charalav.simplerssreader (te
  ▼ res
    ▶ drawable
    ▼ layout
        activity_main.xml
        activity_reader.xml
    ▶ mipmap
    ▶ values
▶ Gradle Scripts
```

Figure 1: The architecture of the RSS reader



5:29

SimpleRSSreader

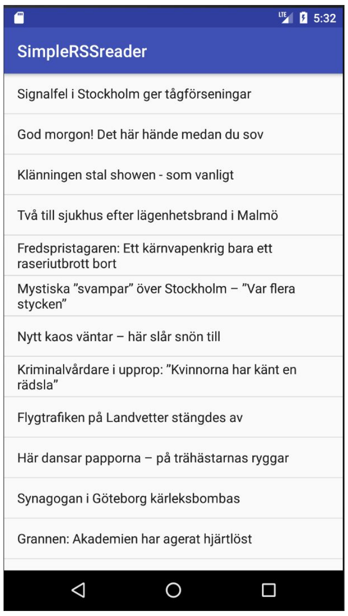
A simple RSS Reader. Just type the url of the feed and press FETCH.

FETCH

An app created for the purposes of ID1212-Network Programming, KTH

Figure 2: The MainActivity screen

- The ReaderActivity is the core of the application (Figure 3). It is the class that communicates with the corresponding server that stores the rss feed. The input of the client was passed from the main actictivity and now it is restored by using a bundle (line 45). The algorithm for fetching all the news is handled by the method rssProcess (line 74). It extends AsyncTask for multithreading reasons, which will be discussed later on the requirements part.


5:32

SimpleRSSreader

Signalfel i Stockholm ger tågförseningar

God morgon! Det här hände medan du sov

Klänningen stal showen - som vanligt

Två till sjukhus efter lägenhetsbrand i Malmö

Fredspristagaren: Ett kärnvapenkrig bara ett raseriutbrott bort

Mystiska "svampar" över Stockholm – "Var flera stycken"

Nytt kaos väntar – här slår snön till

Kriminalvårdare i upprop: "Kvinnorna har känt en rädsla"

Flygtrafiken på Landvetter stängdes av

Här dansar papporna – på trähästarnas ryggar

Synagogan i Göteborg kärleksbombas

Grannen: Akademien har agerat hjärtlöst

Figure 3: The ReaderActivity screen with Aftonbladet rss feed

Considering now each requirement separately:

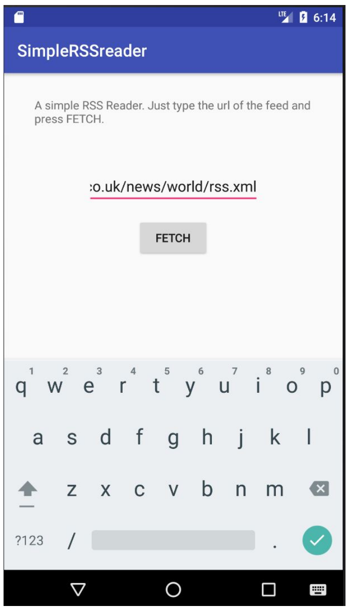
- The client part must run on an android device or an emulator, while the server should be connected to the internet. Server should be able to communicate with the nodes of the application running on the android device. In our case, as mentioned earlier, the android application is running in a virtual android device (Nexus 5x). All the information of the implemented android SDK can be found in the build.gradle file (here). As for the server, no custom implementation was written, but rather all the info that arrive to the client are from web servers from different providers e.g. aftonbladet, bbc, expressen, etc. The crucial part for this android application in order to achieve connection with the server is first to give permission to have access to the internet (mentioned earlier for the manifest file), and a specific line of code (line 68), which opens a connection with the server so that to get the input stream.

- The android app must have a responsive UI. This was achieved by introducing an AsyncTask in the ReaderActivity class (which is the activity that communicates with the server). In this task, 3 methods were override. The first one is called onPreExecute (line 79) and runs on the UI thread. It just displays a dialog for loading context, before the titles of the feed have actually arrived at the screen. The second method is called doInBackground (86) and performs a computation on a background thread. As a result, while processing the rss feed, the client is able to quit the app without having to wait, giving the UI a responsive ability. The third method is called onPostExecute (line 123) and Runs on the UI thread after the doInBackground method. It is used to display the processed list to the user or alert with a warning message if something went wrong.

- The Android app must have a GUI for user interaction developed using UI Views in Android. This was achieved by creating two different .xml files in the layout folder (here), one for each activity. For the MainActivity, a relative layout was used (line 2), while for the ReaderActivity a linear one (line 2).

- The server, if there is a server, must be able to handle multiple clients playing concurrently, which means it must be multithreaded. This requirement was not considered in this assignment, as no custom server was deployed.

- The user interface must be informative. This is clear by considering Figure 4 and Figure 5. The user types a new rss url, then touches the FETCH button, and then he is redirected to the screen with all the rss titles. If he touches one of those he is redirected to this specific one in the browser (Figure 6). If the provided rss url doesn't exist, then a pop up warning message is diplayed to the user (Figure 7).

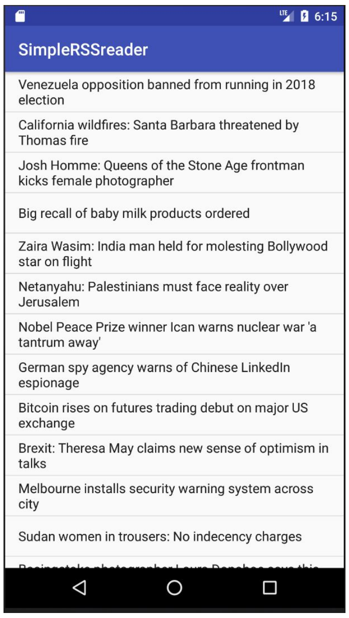


A simple RSS Reader. Just type the url of the feed and press FETCH.

:o.uk/news/world/rss.xml

Figure 4: The entry screen where the user provides a new rss url

The final code of the Java application can be found here. Several comments are included there discussing about essential parts of the program.


SimpleRSSreader

Venezuela opposition banned from running in 2018 election

California wildfires: Santa Barbara threatened by Thomas fire

Josh Homme: Queens of the Stone Age frontman kicks female photographer

Big recall of baby milk products ordered

Zaira Wasim: India man held for molesting Bollywood star on flight

Netanyahu: Palestinians must face reality over Jerusalem

Nobel Peace Prize winner Ican warns nuclear war 'a tantrum away'

German spy agency warns of Chinese LinkedIn espionage

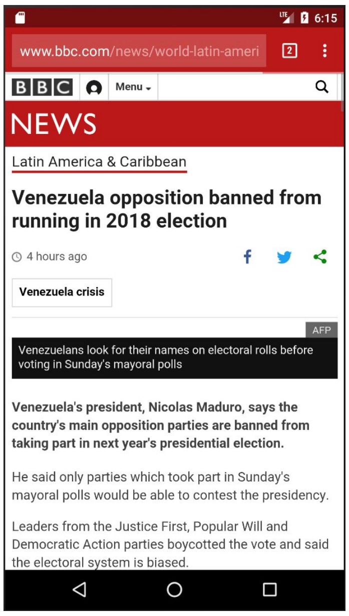
Bitcoin rises on futures trading debut on major US exchange

Brexit: Theresa May claims new sense of optimism in talks

Melbourne installs security warning system across city

Sudan women in trousers: No indecency charges

Figure 5: The second screen with the actual titles of the given feed


www.bbc.com/news/world-latin-ameri

Latin America & Caribbean

Venezuela opposition banned from running in 2018 election

4 hours ago

Venezuela crisis

AFP

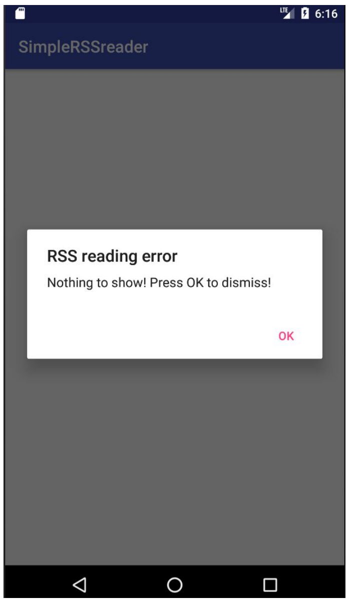
Venezuelans look for their names on electoral rolls before voting in Sunday's mayoral polls

Venezuela's president, Nicolas Maduro, says the country's main opposition parties are banned from taking part in next year's presidential election.

He said only parties which took part in Sunday's mayoral polls would be able to contest the presidency.

Leaders from the Justice First, Popular Will and Democratic Action parties boycotted the vote and said the electoral system is biased.

Figure 6: The new article in the browser when the user touched to see details



RSS reading error

Nothing to show! Press OK to dismiss!

OK

Figure 7: The pop up dialog error

5 Discussion

The main goal of the assignment was to learn about android development.

The requirements of the presented code can be summarized in running an android device/emulator, introducing an internet communication with a server, having a responsive UI for the app, having a GUI for user interaction developed using UI Views in Android, and finally having an informative UI. Any problems that occured during the execution were immediately identified as many 'throw exception' rules were implemented in the Java code.

If something was to be done differently, that would be again as in the previous four assignments, the testing part. It would have been beneficial to perform unit tests (e.g. JUnit), so that to guaranty that everything performs fine without any issues.

6 Comments About the Assignment

This specific assignment was the most interesting from all the previous one. It took me about one day to finish it (with several breaks in between), as I had previously worked with android development.