What is
AndOffline is an open-source (GPL) toolkit that offer some “support” tools for Android phones.
His features takes advantages of existing Android Application and others ones not.
It is written in Java language, so that can run on any operating systyem with a valid installation of a JRE (Java Runtime Environment).

How is composed
The application is composed of two parts, both contained in the downloaded zip file:
- a desktop application to install on your desktop/notebook (the .jar file)
- an Android application to install on the phone (the .apk file)

The desktop part can work without the Android application, but is not true the viceversa.

Desktop application installation
His name is “andOffline-core-x.x-standard.jar” (where x.x is the version number)
To install it right click on it and choose “Open with JRE” or simply “Open” (or similar one, depending on you operating system).
On *nix operating system can be necessary set his execution bit.
Another solution is open a terminal (Ms-Dos prompt on Windows OS) and execute:
java -jar <downloaded-jar-file>.jar

A graphical wizard should be displayed: follow the instructions to proceed with the installation.

At the end of installation, a folder named “AndOffline” will be created (If you use the default name,). It contains all the required dependency.

To run the desktop application execute the file run.sh (on *nix) or run.bat os Windows OS.

Mobile application installation
This part is distributed as .apk file (ie AndOfflineApp.apk) to be installed on an Android phone.
Before install it, you must perform a little setting on phone.
Is necessary enable the deploy of applications not downloaded with the official Android Market (look in the phone settings menu. The exact name can be different between phones and versions).

Connect your phone to the PC with USB cable and copy the .apk file to the sd card. Then you can disconnect the phone from the PC and from the phone go the to the location in the SD card where you have copyed the .apk file, select it to install. At the end of installation the application is available on the phone

Desktop application features details
The desktop application is composed of more tabs panel. Each one provide a different feature, to use them is necessary install some free application on your Android phone.
Is also possible try some features using demo data already provided with the installation (See section “**Use demo data”**).
Let's see in details the single features of each tab:

a) SMS Browser
Offer a more efficient way to view and mange the sms that you have received/sent.
To use it you must install on your Android phone the free application “sms-backup-restore” available on the official Android market.
His official site is: http://android.riteshsahu.com/apps/sms-backup-restore .

After exporting your sms with “sms-backup-restore” you got an .xml file: open it with “Andoffline”.
Now you can filter the sms, execute search or export them is a well formed PDF file ready to print.

Selecting a tree node, appear a menu where is possible use some additional features
Is possible:
- sent by mail the sms text (is necessary confiure AndOffline application to use a mail account to use, see section: “How to configure mail server”)
- sent to Twitter the selected sms text as a “direct message” (ie: an email sent to the email address of a follower, or as Tweet (see section “How to configure Twitter authentication token”).

NOTE: To use this advanced features is necessary configure the AndOffline application to use a valid Email account and a Twitter one. For this purpose use the dedicated “Configuration” menu in the menu bar (See sections “**How to configure mail server**” and “**How to configure Twitter uthentication**”).

Optionally is possible save the loaded messages in a relational database (the supported ones are Sqlite or Mysql: See section **Configuring Database Access** for more info.

b) Call Browser
Offer similar features at the ones of “Sms Browser”, but are applied at the call done/recieved/missed.
Is necessary install on the Android phone the free application “call-logs-backup-restore” on Android market.
The official site is:http://android.riteshsahu.com/apps/call-logs-backup-restore

c) Contact Browser
Offer management features for your contact stored on the phone. To use them is not necessary install third part application on the phone, just export the concacts with the export function of your phone. I should get a .vcf file (only this format is supported), open it with AndOffline Contact Browser.
Note: Only the textual data contained in the vcf file are shown, not other info like photo/pictures associated at each contact.

d) Image encoder/decoder
Premise: it is not a support feature for phone application, but mainly an “add on”.
Allow you to encode/decode data from/to Base64 format (http://en.wikipedia.org/wiki/Base64).
As input you can use a jpeg image (eg myimage.jpg) and convert it to Base64 format (eg myimage.txt) and viceversa.
To see an example of Base64 encoded data try to export your phone contact; you got a .vcf file containing Base64 data representing the image icon associated with a contact.

e) Web server
Premise: to start the web server is required that the system environment variable JAVA_HOME or JRE_HOME are set on your operating system..
On Windows open the Control Panel --> System --> panel “Advanced” --> Environment Variable
On Linu x(Ubuntu and similar) add an entry like this in the file /etc/environment :
export JAVA_HOME=<root-folder-of-jdk-installation>

Where <root-folder-of-jdk-installation> is the root folder thta contains all your java installation.
Can be necessary restart the computer.

AndOffline integrates a server (Apache Tomcat) to allow the management of your database stored JOB from the web (ie using a browser).
Before use this features you must configure a valid database connection with the "Configuration --> Database" menu entry.
To start the server using the start button. Then go to http://localhost:8080/andoffline-web
from your PC or replace "localhost" with an IP remotly reachable to use the application from internet (for example http://10.55.17.62:8080/andoffline-web).
To remotely connect you need to disable your firewall o modify his access rules.

The default credential to access at the application are:
user: andofflineuser
password: pw01

(See: To change default user account see section How to change the default user/password)

The web based application allow to manage from internet the stored jobs.
Obviously you must choose the configured database type that you have configured with the "Configuration--> Database" menu.

Is possible start/stop the server from remote using an sms, no job is required to be configured, simply send an sms with this predefined comand:

execute server start
execute server stop

To do this you must configure the connection between pc and phone.
See section "Phone Manager" and "Sms Command syntax" for more info about phone and PC integration and the syntax to use for sms command to execute stored jobs..

How to change the default user/password
Is possible change the username and/or password used to login in the AndOffline-web application.
Edit the file named "users.properties" placed in the folder:
<base-installation-folder> /AndOffline/web-app/apache-tomcat/conf/andoffline

(<base-installation-folder> is the folder that you have chosen during the installation)

IMPORTANT: respect the syntax when edit it.

f) Phone Manager
Requirements (mandatory): If you don't have it, install the Android sdk on the PC where you connect the phone (See: http://developer.android.com/sdk/index.html).
This is necessary because the monitor running on PC side, uses the "**adb**" command available with the Android sdk.

To connect an Android phone with a PC with USB can be necessary install dedicate driver on PC side (specially if you are using Windows OS). The procedure can be different for each phone.
Try look this for more informations:
http://developer.android.com/sdk/win-usb.html
http://developer.android.com/tools/extras/oem-usb.html
http://developer.android.com/tools/extras/oem-usb.html#InstallingDriver

For other phones if connected with a Linux Os (like my one, Vodafone Huawei Ideos) the are no driver, but only necessary add a configuration file (eg: 51-android.rules) containing a special code vendor-specific (See: http://developer.android.com/tools/device.html#VendorIds) then place that file in the folder /etc/udev/rules.d/ see: **http://developer.android.com/tools/device.html**

To use the Phone Manager features you must follow this step:
1) Install the Android application (andOfflineApp.apk) on phone (see section "**Mobile application installation"**) and start it.
2) Connect the phone with PC with USB cable (Before do that, see above mandatory requirements above)
3) On the phone enable the "USB debug" (look on the "settings" menu of the phone, the exact name can vary from different phone) You should see a "bug icon" on the phone status bar.
4) On PC, look at the file "pc-listener-service.script" under the "doc" folder of you AndOffline installation.
This file contains a command to be executed in a shell/console of you operating system to start the handler of the messages written by the andOfflineApp running on the phone.
The shell/console must be opened in the same folder where the executable of "**adb**" command is placed
For example, if "adb" is placed in the folder /home/john/android-sdk/platform-tools you must open a console/command prompt in this location and execute the script provided in file " pc-listener-service.script".
(If the "adb" command is in the PATH environment variable you can run the script in any writable folder).

For example for *nix OS the adb is placed in the folder
<installation-folder-of-sdk>/adt-bundle-linux-x86-20130514/sdk/platform-tools

```
IMPORTANT: don't close the console/prompt where you run the script!!
```

The script create a file named **`andofflinePhone.txt`** in the same folder where you run it.

On the "Phone Manager" tab of AndOffline follow this step:
5) Set the "swap folder": this is the folder wher the commnad execute at point 4 has created a file (ie: **`andofflinePhone.txt`** looking at the sample in "pc-listener-service.script" file)
6) Adjust the polling frequency: this parameter indicates the width of the interval between to checks of the monitored file (eg: **logout.txt**). The value to set, depends on the receiving frequency, don't set a too little value.
7) Start the listener using the dedicated button on the panel.
8) Start the AndOffline application on the phone, set the sms prefix and start the receiver

- At this point the desktop application is ready to receive incoming SMS from the connected phone to execute a database stored job (See section "**SMS command sintax**" for information about the incoming sms sintax).

In the "Monitoring Panel" is possible check the real-time situation about the total amount of received SMS processed and the ones in queue.

IMPORTANT: at each exection is necessary delete the old logout.txt file or redirect the log

message to another file. This is necessary to prevent another processing on the same file.

--> To Stop the monitoring:
1) stop the receiver on the phone applicaton
2) stop the listener using the dedicated button on the panel.
3) close the console where you have launched the script with "adb"
4) disconnect the phone from the PC (using the safe mode that disabel USB .....)

The file **andofflinePhone.txt** can be deleted (will be recreated next execution)

g) Job Manager
Allow you to view add and edit the jobs store on a previosly configured database (See section "**Configuring Database Access"**).
That jobs are remotely excutable with an sms (NO internet connection is required).
A "job" is composed of this fields:
id: a numeric value that identify a job. User can't decide it, is generated in automatic by the database in phase of insertion..
name: a user defined name of the job (eg. "temperature reader job")
interpreter: the language interpreter of the script.
It can simply be the executable of the script interpreter. (eg: java, pyhton) in case of it is configured in your operating system PATH environment, or the full path to it (eg: /bin/sh, /usr/bin/python, /home/john/programs/jdk/bin/java).
InterpreterAgument: the arguments to pass at the above interpreters.
For example, if you want execute a java ".jar" archive, the interpreter is "java" and the 'InterpreterArgument' is "-jar"
script: fully qualified path name to a script/file to pass at the interpreter to be executed
scriptArgument: optional input arguments for the "script" (they are like the ones that you pass from the command line) . Don't confuse the argument for the script with the ones for the interpreter.
description: a user defined description for the job (eg: a little description about what the job do)
last execution date: the date when the sms command is received and then the last the job execution. It is automatically updated at each job execution
Executor Msisdn: the last phone number that has sent the sms command to execute the job.

Example 1:
Suppose that you want execute a python script named "myscript.py" placed in the folder /home/john/test (for windows C:\program\john\test) and your python interpreter is in the path /opt/software/python (for Windows OS could be :C:\programs\python\python.exe).
That script accept as input argument a person name and surname (eg: john doe)
In this case:

name: This is myscript python job (or another one)
interpreter: /opt/software/python (C:\programs\python\python.exe for Windows)
In case of "python" command is configured in OS PATH variable will by
interpreter: python
interpreterArgument: none, this is an optional field (not all the interpreters have argument).
script: /home/john/test/myscript.py
scriptArgument: john doe
description: Execute myscript.py python scipt (or another one)

Example 2:

You want execute a java jar archive, for example 'Archive.jar' placed in the folder
' /home/john/programs/'
(ie: the “jar” archive executed with: java -jar nameOfArchive.jar)
In this case:
interpreter: /home/john/Programmi/jdk1.7.0_21/bin/java (or simply j'ava' if configured in system PATH environment).
interpreterArgument: -jar
script: /home/john/programs/myArchive.jar

Example 3:
On Linux to execute a .sh script placed in '/home/john/programs' should be:
interpreter: /bin/bash
script: /home/john/programs/myScript.sh
scriptArgument: arg1 arg2

Important: leave blank the 'Interpreter' field if you want execute a “.bat” script on Windows.

NOTE: is possible have two identical job but executed with different “argument” values
(one with “john doe” and the other one with “bill smith”)

SMS command syntax
This section describe the syntax that an incoming sms (command SMS) must respect to be correctly processed by the receiver on PC side.
Currently, there is no Android client application that help the user to compose the command sms (maybe someone can it :)).
So that you must care to respect the right syntax .
(tip: you can export the jobs list to a pdf file using the dedicated function)

PREMISE: all the incoming sms MUST start with a pre-defined prefix (you can chose on the AndOffline phone application), so that the AndOfflineApp running on the phone can understant that the incoming sms must be catched and processed.
Without that prefix a message will not handled, because will be considered as a “normal” one instead of a command one.

The general syntax is for command sms is:

<prefix>space<command>space<predicate>space<argument>

Where:
<prefix> the prefix that all the command sms must have at the beginning
<command> must be “execute”
<predicate> can be “server” or “job”
<argument> can be “start” or “stop” in case of <predicate> is “server”; otherwise if <predicate> is “job” is a number representing the id of the job to execute.
The “id” of a job can be obtained looking at the jobs listed in “Job Manager” panel.

Example of sms command (the prefix is omitted for simplicity):
- Execute an existing job stored on the DB whose id is '1'
  execute job 1

- Start the embedded web server server
  execute server start

- Stop the embedded server
  execute server stop

In the current version there are no features that allow to know if a job was executed successfully or has finished (eg: like a response sms) (There is only a log file)

Configuring Database Access

Note: if you plain to use only sms, call and vcf browsing functions, is not necessary configure a database, so that you can skip this section.
After the creation of a database configuration is possible save sms, call or job in a relational database (Sqlite or Mysql).
(note: using a Sqlite database, the information are stored on a filesystem file of your computer, this is the simplest soution for most of cases, and don't require to install other software like Mysql database).
The database configuration is done only the first time, unless you want change it in future.
To configure it use menu entry named “Configuration --> Database”.

The step to follow are:
1) Configure and save the connection parameters for a DB connection
Configuration ---> Database ---> Create Configuration

2) Create the database structure (ie tables and schema) using the configuration parameters previously stored.
Configuration ---> Database ---> Create Schema

If you want use an existing databse containing the reuired structure, skip this step.

To load the previously created configuration and update it:
Configuration ---> Database ---> Edit Configuration
If you change the database name can be necessary re-exectuethe step 2.

How to configure Twitter authentication

This configuration is required ONLY if you want sent by Twitter an sms text (this is an addo-on features, no a “core” one).
Go to the “**Configuration ---> Twitter**” menu and fill the input form.
The required parameters must be obtained from Twitter website registering “Andoffline” application as authorized to call Twitter.
Obviously to do this is necessary that you have a valid Twitter account.
Here the base steps:

1) Register the application (ie: AndOffline) at https://twitter.com/oauth_clients/new
using your Twitter account to acquire a 'consumer key', and 'consumer secret' in advance
Remember that the registered application MUSTN'T be set as “read-only “.

2) follow the steps provided by Twitter official guide. At the end you'll get the "OAuth settings", that are:
- Consumer key
- Consumer secret
- the access token to sign requests with your own Twitter account

Insert them in the AndOffline form and press “Save”, the configuration is finished.
Please, refer to official Twitter website for more detailed informations.

How to configure the smtp mail server
This configuration is required ONLY if you want sent by mail an sms text currently showed in the sms tree panel (like the Twitter features, this is an add-on features, not a "core" one, so that you can skip it).
Go to the "**Configuration**" menu entry and provide the required fields (the input form already contains some sample parameters for a gmail account).

Use demo data
If you haven't installed the Android applications "call-logs-backup-restore" nor "sms-backup-restore" on your phone is possible try the application features using some fake data.
Simply open the files placed in the "/example" folder under your installation one.
The file are named "calls-sample.xml","sms-sample.xml","contact-sample.vcf".

Debug and monitoring
Many of the operations performed by AndOffline gui application are traced in a log file named "andoffline.log" and placed in the "log" folder of your installation.
To understand in more details some error messages of the gui you can look at this file.

The operations, errors etc executed with the web console are logged in a file named "andoffline-web.log" placed in the folder "/web-app/apache-tomcat/logs" under the main installation folder.

Testing information:
Android Application tested using Vodafone Ideos with Android 2.2 Froyo.

For developers:
Requirements: Apache Maven, Eclipse ide, jdk 1.6 (or greater), Android SDK + Eclipse ADT bundle
After the checkout of the code, execute:
mvn eclipse:clean eclipse:eclipse
in the folder where is placed the root pom.xml (ie the one in the folder andoflfine-suite).
Now you can import the project in Eclipse (File- import- Existing project in workspace).
The suite is composed of three project:
- AndOffline: the core part, is java based desktop application
- AndOffline-web: the web application that allowed manage from the the browser the stored job
- AndOfflineApp: the Android application (import it directly with Eclipse ADT bundle).

To buidl the finaj jar (the one with the installer....) execute:
mvn clean package
in the folder where is placed the parent pom.xml (where you have executed mvn eclipse:clean eclipse:eclipse) The final jar will be placed in the folder named "target" of "AndOffline" project.

Know bugs:
Some little layout problems on Windows OS.

License
The AndOffline application is licensed under the term of GPL V3 license. For the third part library used see "Help" –> "About" menu and their website for more information.

Android is registered trademark of Google Inc