Cura document

1. Set up server
2. Download these suggested software (below if you don't have it):
   * **MySQL (server and workbench)**

http://dev.mysql.com/downloads/installer/

* + **Apache Tomcat 7**

https://tomcat.apache.org/download-70.cgi

* + **JRE and JDK 1.7**

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

* + **Apache Maven 3.3**

https://maven.apache.org/download.cgi

1. Open MySQL workbench
   * Create a new connection. Set everything as default.
   * The connection's username and password should both be "root".
2. In MySQL workbench, run the script files located in com.kms.cura.dal\resources\SQL Script:
   * Make sure connect to the connection that you've just created in step 2 first then File/Run SQL Script and navigate to the SQL scripts.
   * **The scripts must be run in specific order**:
     + InitDatabase
     + InitConditions, InitSymptom, InitInsurance, InitMessage, Password\_Code, AddWorkingHour
     + InitApptNotification, InitMsgNotification, Conditions\_Symptoms, Insurances\_Facilities, Patient\_Health, Specialities\_Conditions, Appointment
   * NOTE: Make sure you set character set to **UTF8** before running to be able to encode Vietnamese characters.
3. Go to cura main folder and open cmd there
   * Run "mvn package" or "mvn clean install" and wait for the web app to be compiled
     + “mvn clean install” will auto replace the old *war* file
   * After the compiling process is done, go to com.kms.cura\_server\target:
   * There should be a generated file calls "com.kms.cura\_server-0.1.war" there.   
     Copy it to your Apache Tomcat's "webapps" folder. Rename to whatever name you set for link to the server.
4. Open cmd in Apache Tomcat's "bin" folder, and run "startup.bat" or run directly the batch file “startup”
5. Once the server done starting up, you should be able to access it normal.
   * Check localhost:8080/com.kms.cura\_server-0.1/cura\_server/hello to make sure if it's working correctly (replace "com.kms.cura\_server-0.1" with a new name if you change the war file's name in step 6)
   * Change “localhost” with the server address if you deploy the app on a server computer, the current server for Cura is “192.168.74.141”
6. Set up app:

* Build apk file in cmd: navigate to the root of your project directory
  + On Windows platforms, type this command:

> gradlew.bat assembleDebug

or

> gradlew.bat assembleRelease

* + On Mac OS and Linux platforms, type these commands:

$ chmod +x gradlew  
$ ./gradlew assembleDebug

or

$ ./gradlew assembleRelease

* + - The first command (chmod) adds the execution permission to the Gradle wrapper script and is only necessary **the first time** you build this project from the command line.
  + After you build the project, the output APK for the app module is located in **app/build/outputs/apk/**
  + Each time you change a source file or resource, you must run Gradle again in order to package up the latest version of the application.
* Running on emulator:
  + Open the AVD Manager and launch a virtual device
  + In the Virtual Devices view, select an AVD and click Start.
    - For Cura, to support Google cloud service, you should create a device with **API 22**, the api after that doesn’t support GCM yet.
* Install your application on emulator:
  + From your SDK's tools/ directory, install the .apk on the emulator:  
    > adb install <path\_to\_your\_bin>.apk
    - Your .apk file (signed with either a release or debug key) is in your module build/ directory after you build your application.
  + If there is more than one emulator running, you must specify the emulator upon which to install the application, by its serial number, with the -soption. For example:
    - adb -s emulator-5554 install path/to/your/app.apk
  + To see a list of available device serial numbers, execute   
    > adb devices
    - If you don't see your application on the emulator, try closing the emulator and launching the virtual device again from the AVD Manager.
* Running on a Device:
  + Enable **USB debugging** on your device. You can find the option under **Settings > Developer options**.
    - **Note:** On Android 4.2 and newer, **Developer options** is hidden by default. To make it available, go to **Settings > About phone** and tap **Build number** seven times. Return to the previous screen to find **Developer options**.
  + Ensure that your development computer can detect your device when connected via USB
  + Once your device is set up and connected via USB, navigate to your SDK's platform-tools/ directory and install the .apk on the device:  
    > adb -d install path/to/your/app.apk
    - The -d flag specifies that you want to use the attached device (in case you also have an emulator running)
* Using Genymotion:
  + Genymotion, for default, do not support functions related to Google service (map, GCM)
  + To use Genymotion, visit <http://stackoverflow.com/questions/20121883/how-to-install-google-play-services-in-a-genymotion-vm-with-no-drag-and-drop-su> to install Google play on specific Genymotion emulator