**Backend App Management System:**

Source code is under rufuslabs\_update\_service/:

1. ApkDBHelper.js
2. ApkFileManager.js
3. ApkParser.js
4. AppsUpdateService.js

Programming Language:

Javascript

Dependencies:

NodeJs (with modules: multer, admzip, express,path,jade)

Mysql

Platform:

Windows or Mac OS or Linux

Setting up:

1. install mysql
2. create database named : update\_service\_database.
3. Enter username and password of the database at the lines 7,8 of file: ApkDBHelper.js
4. Install nodejs
5. Install nodejs packages using commands in terminal:
   1. npm install express
   2. npm install multer
   3. npm install adm-zip
   4. npm install path
   5. npm install jade
   6. npm install @mysql/xdevapi
   7. npm install eval
6. Find your ip address of your server/pc , for example 192.168.1.8.
7. Enter the ip address at line 137 of the file: AppsUpdateService.js
8. Follow the instruction at line 11 of the file ApkParser.js.
9. In terminal, go into directory rufuslabs\_update\_service/ and enter “node AppsUpdateService.js to start the service.
10. Access the management web page in your browser at : <http://192.168.1.8:8081> (if you set the port as 8081)
11. About how to operate functions in the web page, please check the video at: https://youtu.be/bewEpzmhTig

Notes:

1. information is stored in two tables in mysql database: version\_info and app\_info.
2. Apks are uploaded into folder: public/uploads/<uploaded time>/\*.apk.

**Android update service app:**

Source code is under rufuslabs\_update\_service/client/CuffUpdateService. This is an Android project.

Platforms:

Android 4.4.4

Setting up:

1. Change the ip address and port at the line 25 of the file AppsUpdateBackground.java to the address and port of the web service, which is, by the example mentioned above, 192.168.1.8:8081
2. Run the application and operate the functionality by the video at:

https://youtu.be/Y1MFdqWG4wQ

Note:

1. minimum user interface is implemented. You guys may want to change it later.
2. Planned to make the installation run silently in the background. However, right now I can only install the downloaded application by calling Android apk installer activity which require installation show in front due to limit of privilege of moto g. But the goal can achieve on the board by calling a reflection method, after we put service into system app folder and sign it with system certificate. About this method, there is more information online when you google it.
3. This android background update service is programmed to request update information at certain interval of time (about 4 hours). I also made an activity to arouse the update service by clicking the button, which helps debugging and testing. The activity is the interface you will first see when you open this application.