Overview

Android Performance Evaluator is a service that allows developers how efficiently their application is running on the android OS.

This product is not complete. All of the wording will need to be revised several times before it is finalized. The graphics and layout of the screens is shown here are not the final product. The actual look and feel will be developed over time with different input from product managers and programmers.

Installation Instructions

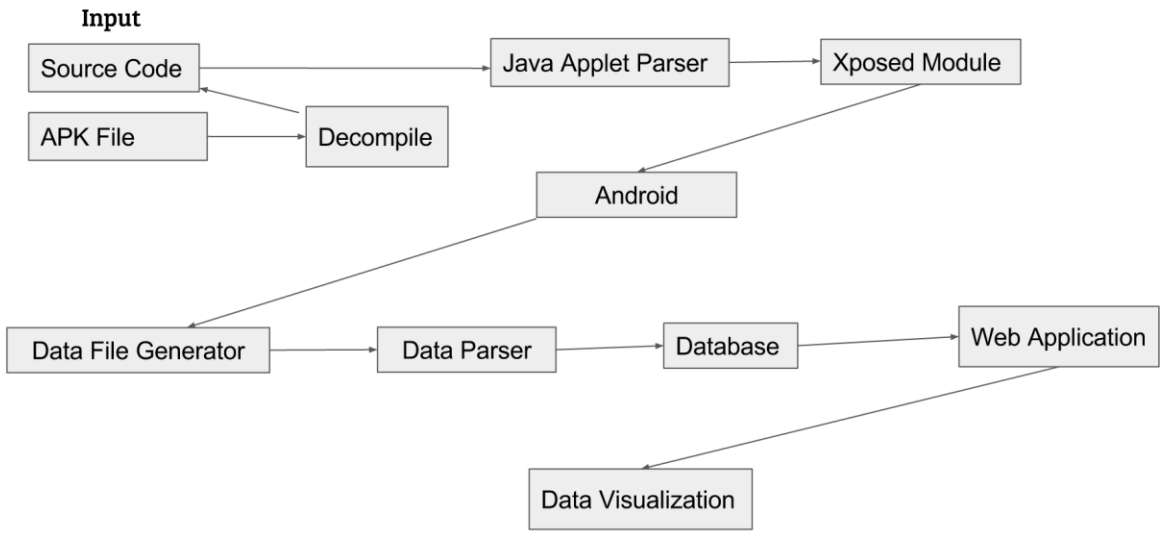
A brief high level "how-to-use" guide for the current build.

Last revised 3/30/2016

1. Start the Java app. It will open the file explorer. Select the source code you wish to parse or the apk you wish to analyze. Select the entire package if passing in source code - DO NOT navigate to the /src folder or any other subdirectory. The parser needs to the entire project structure to generate the parsed output.
2. After selecting the code/apk to analyze, select Parse. Parser will parse through your uploaded code (and decompile it if necessary). It will then produce an output file that ModuleBuilder will analyze in order to generate your Xposed module.
3. When the status on the file explorer states that the module has been built, close the file explorer. The module file (moduleFile.java) will be ready for installation.
4. Install moduleFile.java on your Android phone or emulator.
    a. Currently, Android Studio installs the module for us. Eventually, the installation will be automated.
5. Once the module is installed, restart your phone or emulator to complete the installation. You are now ready to analyze the app.
6. Open a terminal and navigate to where your adb resides (likely android-sdks/platform-tools).
7. Run "$ ./adb logcat -s Xposed". You should see the module hook onto the package you are analyzing and produce the method start and end times.
8. Play with the app to produce more output for the logcat. It will record the method times related to the buttons you select.
9. Copy the output to data.txt and upload to dataBaseListener. dataBaseListener will parse this output and upload it to the database.
10. Log onto http://localhost/softDev/index.php. Username is username, password is password.
11. Select Projects and then the app you wish to analyze. Charts will appear with your uploaded data. Select Make CSV or Make PDF to get the stats as raw data.

Data Flow

Below is the data flow of the program and should give a general sense of how to use the Android Performance Evaluator. Details of each component will be described in detail later.



Technologies Used

Below are the technologies currently used in Android Performance Evaluator.

Parser Components

The Java Applet Parser takes in either source code or an apk as input. If the input is an apk, it will decompile said apk and parse its source code. An apk or source code directory can be selected through the file explorer after hitting the “select file” button as shown in Figure 1.

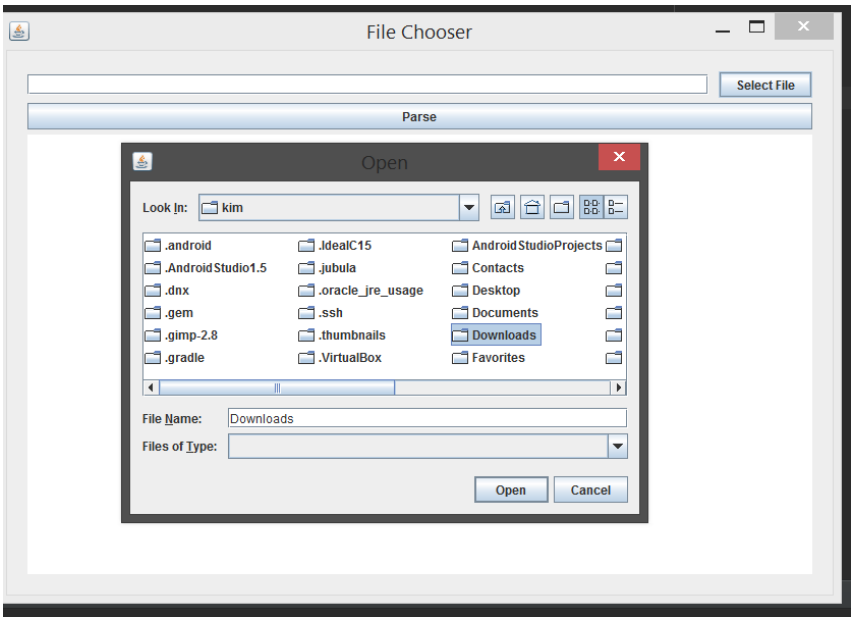


Fig 1) File Explorer in choosing the apk or source code

Once the desired file has been selected, then it will begin the automatic parsing process once pushing “parse” as shown in Figure 2. Java Applet Parser will pull the package name, methods, parent classes, parameters, etc. for each method and will output that into a source file to be parsed by ModuleBuilder.

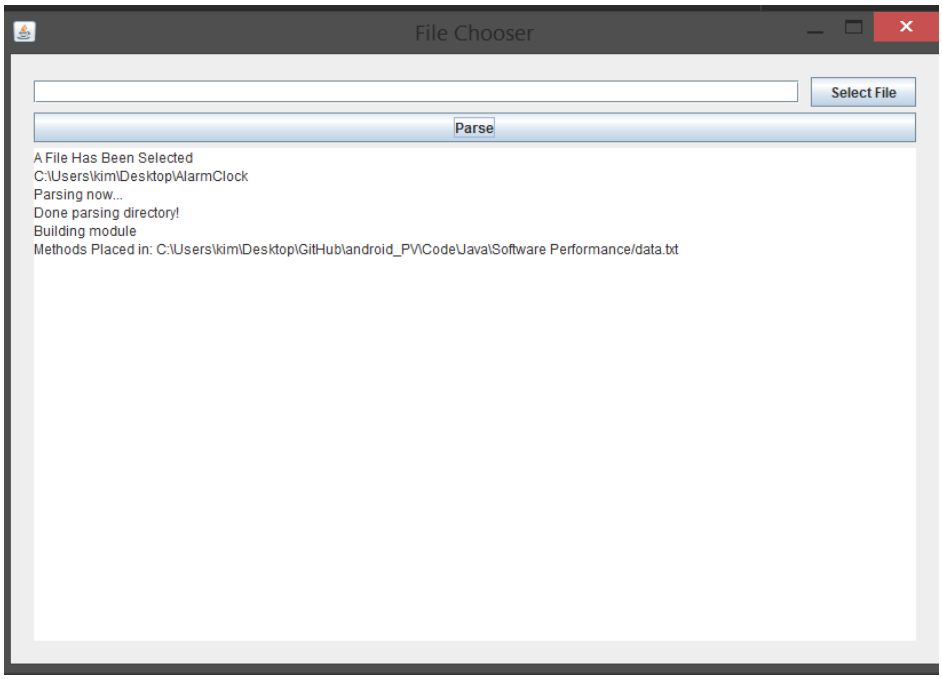


Fig 2) Parsing demonstration with output from the terminal

Android Components

The ModuleBuilder from the Java Applet Parser will be then generate an Xposed module as shown in Figure 3.

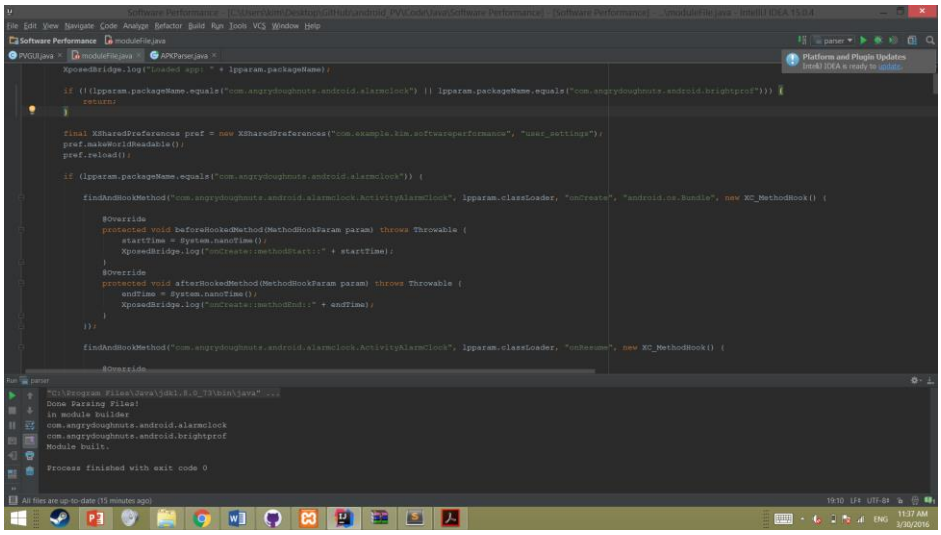


Figure 3) Xposed Module code

The Xposed module hooks methods based on the app you are analyzing. When a hooked method is executed, Xposed will record the start and exit time of the method in order to calculate the full execution time. All the user has to do is install the Xposed module on his device (see Installation Instructions) and use the app to be analyzed to collect data. Once the user is done with collecting data, there will be a button on the android screen which will save the logcat. From this point on, the logcat be manually moved over. (see Installation Instructions).

Web Application Components

The dataBaseListener, upon upload of this output, will add it to our database to be represented on the website, and organize it to the required tables. From this point, the user will move onto the website application. First, the user will be prompted to login to authenticate the user's application and traces as shown in Figure 4.

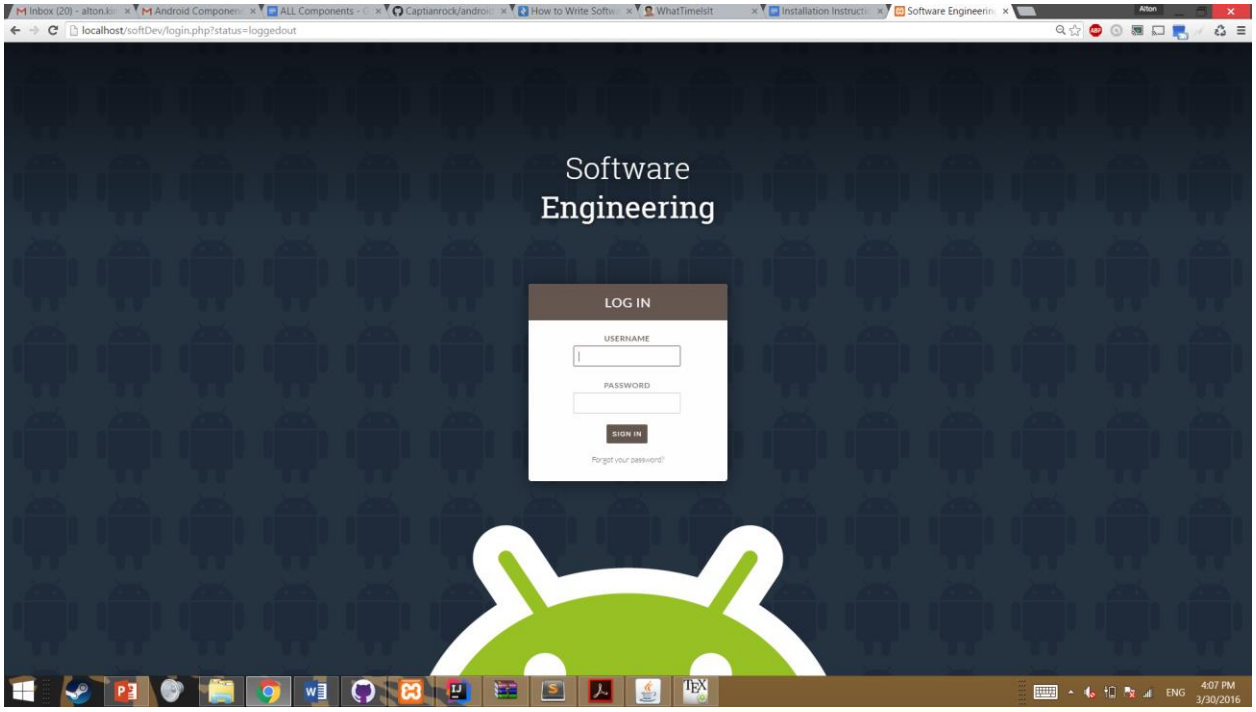


Figure 4) Login Screen for the User

This will guide the user to the homepage screen where they have the option to navigate through their projects, to log out, make a csv file, or make a pdf file of the raw data as shown in Figure 5.

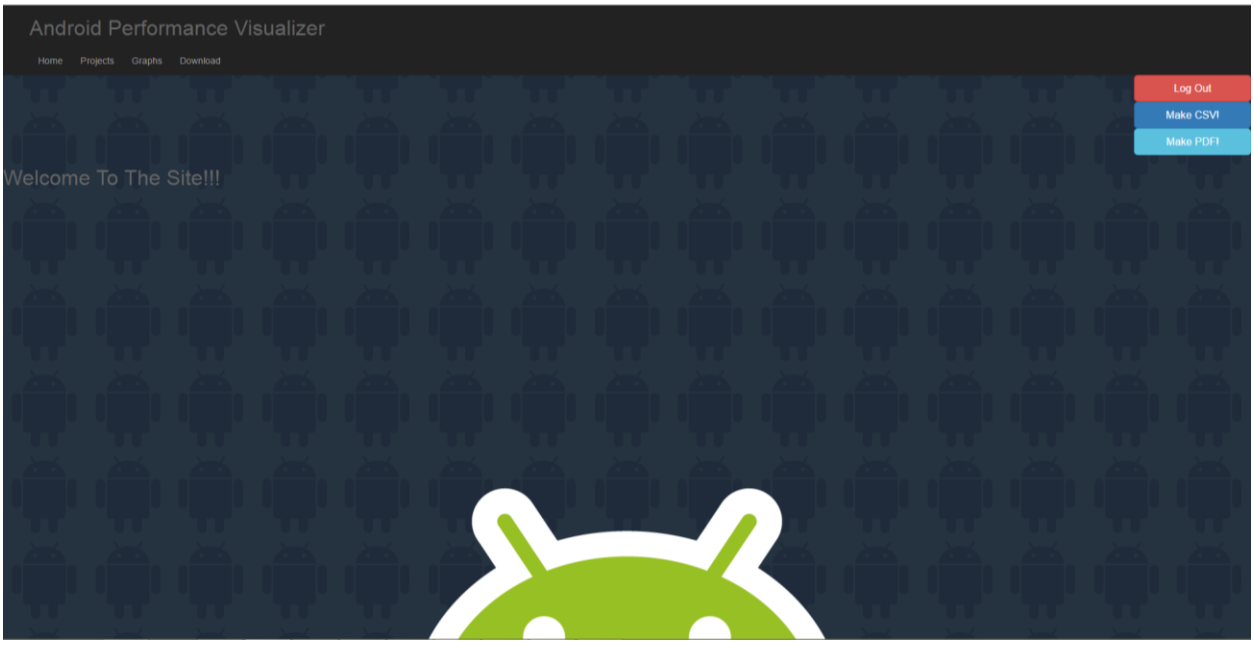


Figure 5) Home page with options for the user.

When choosing the option to see their projects they will be able to have access to the graphs that visualize the data as shown in Figure 6. The user can play around with the graphs and at the bottom be able to see certain statistics.

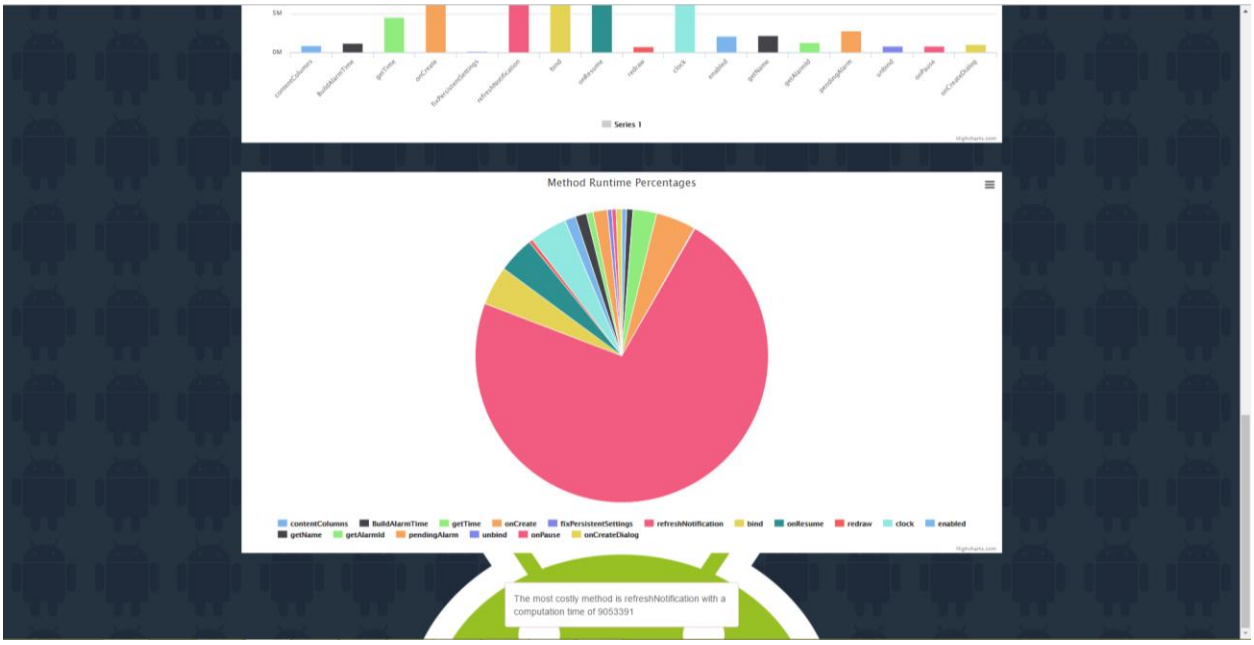
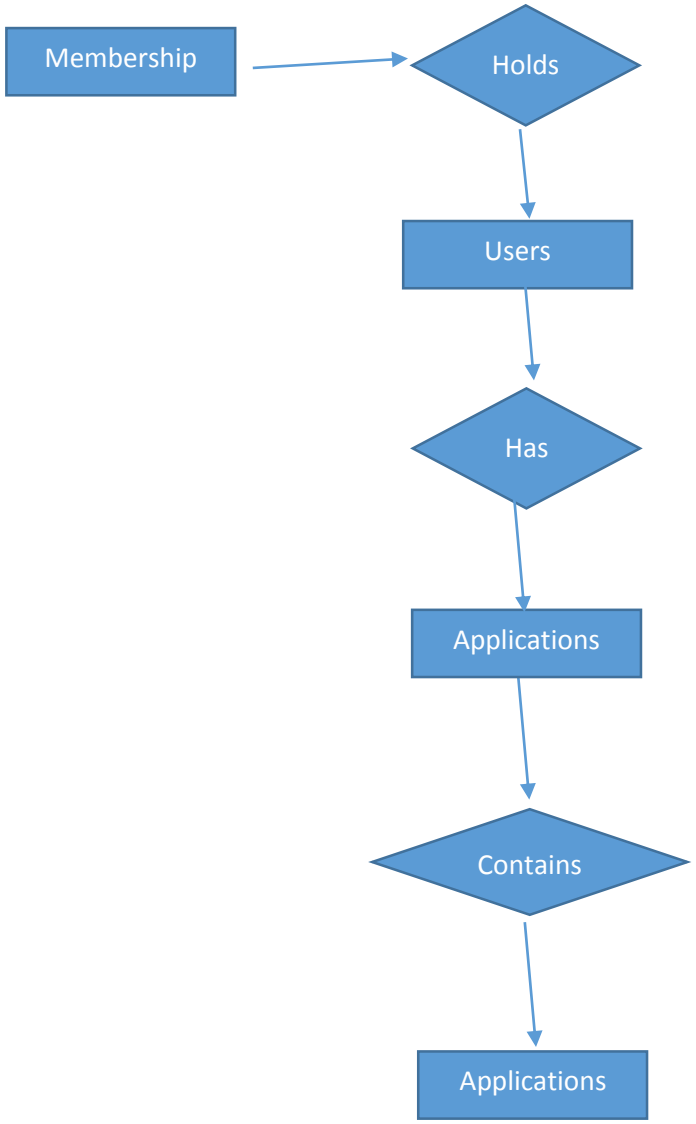


Figure 6) Example of Data Visualization

Database Model

Here is the current planned layout of the database. Each user logged in will be have their own set of applications and traces that are accessible to only them.



Entites:

Membership(<u>username</u>, id, password)

Users(<u>username,</u> application_name)

Applications(<u>application_name,</u> traceId)

Traces(<u>traceId,</u> methodName, startTime,endTime