



Juzz Say: A Voice Enabled Android Application using Python

Ray Subharthy

Academy of Technology, Aedconagar, Adisaptagram, West Bengal, INDIA

Available online at: www.isca.in, www.isca.me

Received 31st August 2014, revised 18th September 2014, accepted 15th October 2014

Abstract

This research paper presents a brief overview of the android application “Juzz Say” developed using Python that is an application with some voice enable service. It has five voice enable services, SMS, Camera, Music, SOS, Calorie-Fat Calculation. A database is maintained for each service. Total application is voice enabled except calorie calculation. At the first use of any service will lead the user to the settings. After finishing the settings he or she can use that application. If needed the user can set all the settings later on to select the ‘Settings’ option. This project is built using Python language and in the SL4A (Scripting Layer for Android) environment that provides several types of APIs to use the functionalities of an android phone. The Scripting Layer for Android is a library, allows the creation and running the scripts written in various languages directly on Android devices. The Database of this project is built using SQLITE3, and some pages are made of HTML. This project has been made to reduce the complexity of using Android services. We can use the services while working. We do not need to touch our phone to use those services. This application is also fruitful for those people who are blind and still have the interest to use android phone.

Keywords: Android application, voice enabled application, Python application, SL4A.

Introduction

Android OS: The Android is an Operating System acts like a software stack of different layers. Each layer is a group of several program components. It includes operating system, and important applications. Each layer in the architecture provides different services to the layer.

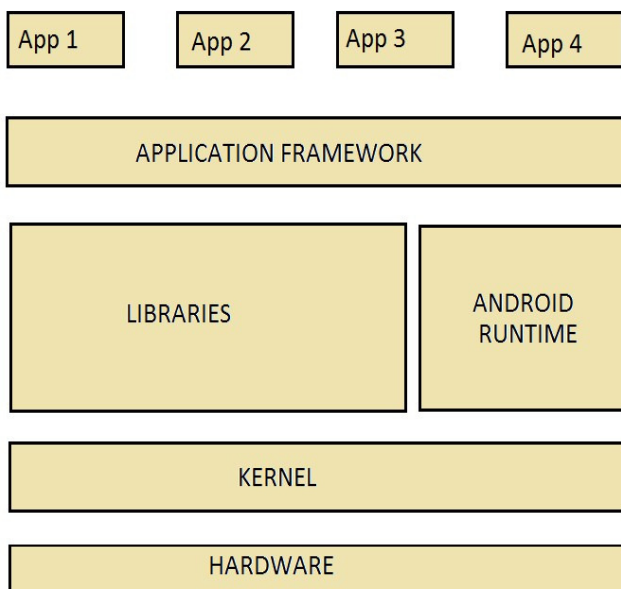


Figure-1
Architecture of android OS

Android uses the Linux for all its core functionality. Android is built on a most popular platform; it made the porting of Android to variety of hardware, a relatively painless task. The next layer is the Android’s native libraries. It is this layer that enables the device to handle different types of data. These libraries have written in C or C++ language for a particular hardware.

Python: Python is languages that nowadays widely use. It is a high-level programming language. It is language that is object oriented as well as procedural so that we can do both types of program. Python provides constructs to enable programs in a small and large way. Python supports multiple programming, object-oriented, imperative and functional programming. It is a dynamic system. Python interpreters are available for many operating systems. As other dynamic languages, we can use Python, but it also is used in a wide range of non-scripting. We can use third-party tools such as Py2exe, Pyinstaller. Python code can be packaged into executable programs¹.

PY4A: Python For Android is Python built to run on Android devices. It is made to be used together with SL4A. Nearly all the actual non-python specific documentation can be found at android-scripting. To be able to run Python on android, you need to compile it for android. And you need to compile all the libraries you want for android too. Since Python is a language, not a toolkit, you cannot draw any user interface with it. The Python for android project includes a Python module called “android”. It consists of multiple parts which are mostly there to facilitate the use of the Java API².

SL4A: The Scripting Layer for Android is a library that looks after the creation and running of scripts written in various scripting languages. SL4A gives us the power to use to make or to automate our Android device in a different way. SL4A currently supports JRuby, Perl, PHP, Python, and Rhino. SL4A provides an interactive console where we can write a line of code and immediately see the result. SL4A makes it possible, in some cases, to recycle code we have written for a desktop environment. SL4A makes it possible to write code for Android devices other than Java and in a more optimistic way³.

The **web View Show** API makes it possible to open a custom HTML page specified by the URL (it's possible to use file:///) in a Web View. Web View offer the same familiar Android class available to all supported languages.

SL4A and Python for Android installation: Installing SL4A and py4a on an Android device is similar to the process for any other Android application. You can scan the QR code on the main SL4A project site with your android device to download the SL4A and PY4A .APK file. It automatically starts when the download is finished⁶.

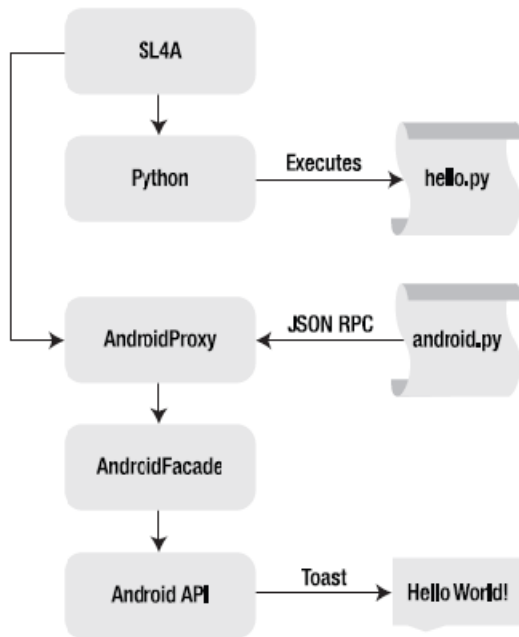


Figure-2
 Flow Diagram of SL4A

Sqlite3: SQLite3 is a python library that provides a lightweight database that doesn't require a different server process and allows it accessing the database using a variant of the SQL. There are so many applications can use SQLite3 for internal data storage. It's also possible to make the prototype of an application using SQLite and then change the code to the larger database like Oracle. The module was written by Gerhard Haring. It provides a SQL interface with the DB-API 2.0 specification described by PEP 249. To use the module, you must first create a Connection object that represents the database⁴.

HTML: Hypertext Markup Language is a system that is standardized for tagging text files. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets. The first tag in a pair is the start tag, and the second tag is the **end tag** (they are also called opening tags and closing tags). A web browser can read HTML files and make them a visible or audible web page. The browser uses them to interpret the content of the page. HTML tells the structure of a website for presentation and making a markup language, a programming language⁵.

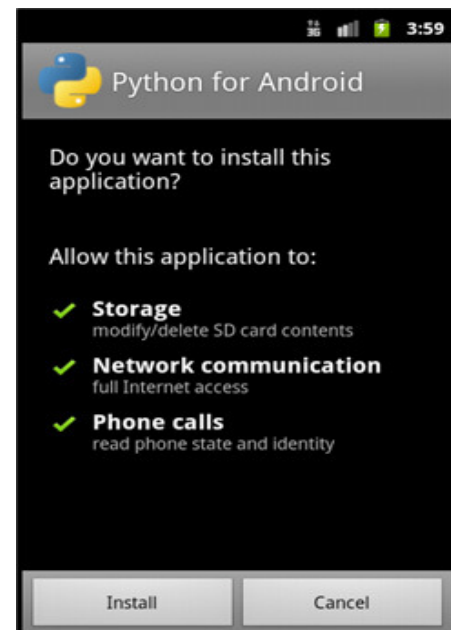
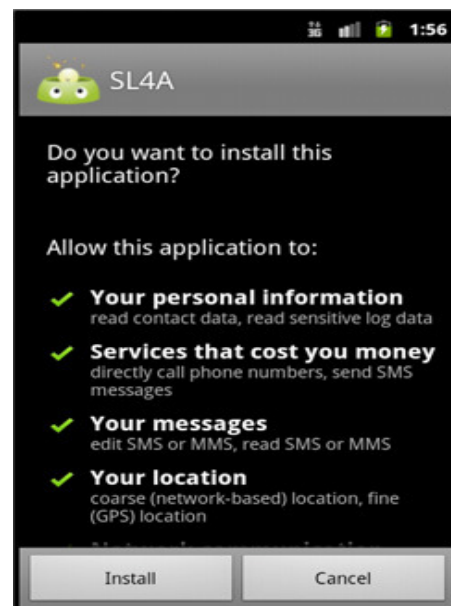


Figure-3
 SL4A and Python for Android Installation page

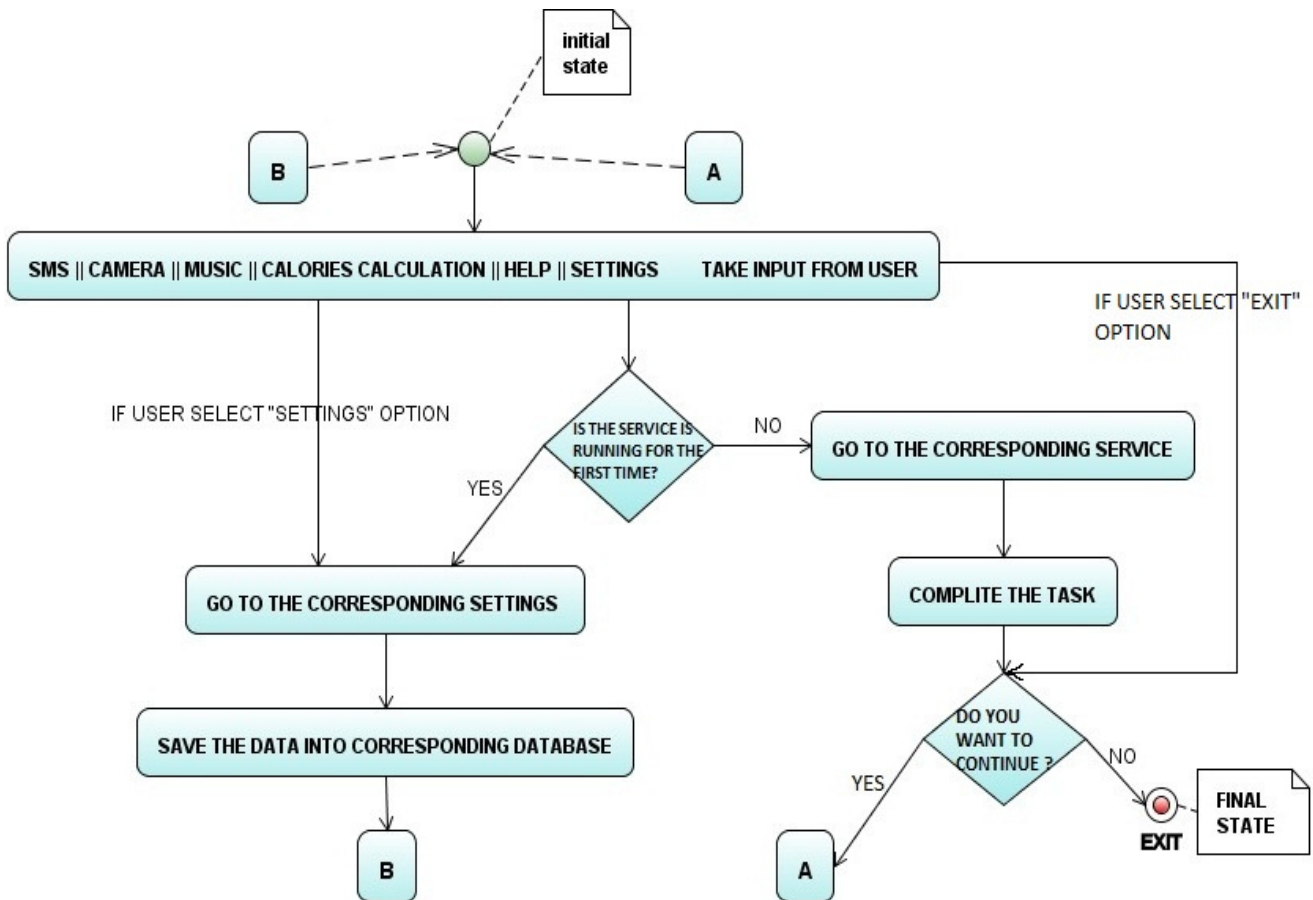


Figure-4
 Flow Diagram of the application “Juzz Say”

Methodology

In this section, I have described the methodology I have used to achieve what I have proposed. There is, some figures related to the data structure I have used and the flow diagrams, along with the step algorithms used.

Database Structure: I have used sqlite3 for creating database. This application consists of i. cur. Execute ("create table key (id varchar(10),keyword varchar(20))") Table 'key' is used to store the user keywords for all the services. ii. cur.execute("create table help(ph_no varchar(12))") Table 'help' is used to store the phone numbers that the user wants to call for help. iii. cur.execute("create table calories(Date varchar(30), calories varchar(20), fat_reductionvarchar(20), speed_in_km/hr varchar(10))")

Step algorithm:

INPUT: voice input of a service name from user.
OUTPUT: user will use the service.
STEP 1: Ask the user to select a service via tts Speak () method and take input from user by recognize Speech () method.

- STEP 2:** IF the requested service is running for the first time then system will redirected to the settings page, where user have to set the keywords for the corresponding service.
- STEP 3:** Save the keywords into database. Then go to STEP 5.
- STEP 4:** ELSE perform the requested task.
- STEP 5:** Ask to user “do you want to continue?”, if yes than go to STEP 1. Else, Exit.
- STEP 6:** In the STEP 1, if user will choose “Exit”, then go to STEP 5.
- STEP 7:** In the STEP 1, if user will choose “settings”, then user will be sent to the setting page, and will go to STEP 3.
- STEP 8:** STOP

Results and Discussion

In this section, I have given some screen shots of the application I have designed. The maximum part of this application is voice enabled that’s why very few amount of UI is present in this application. I have designed some html page in the calories calculation part.

When user is about to give any input then he/she will see the following screen.

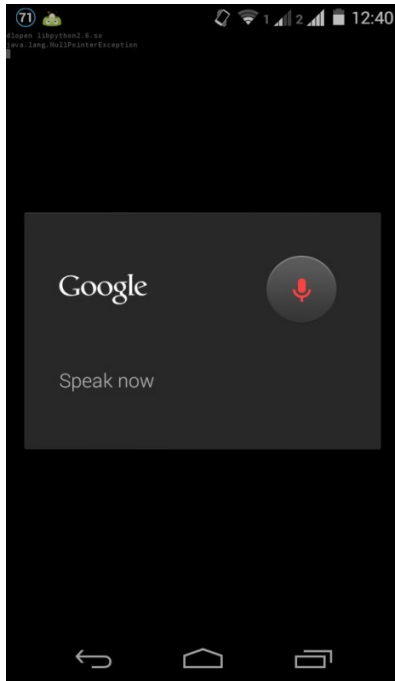


Figure- 5
 Taking user voice input



Figure-7
 List of the evaluation of user's calories

When user wants to start calories calculation service then he/she will see the following screen

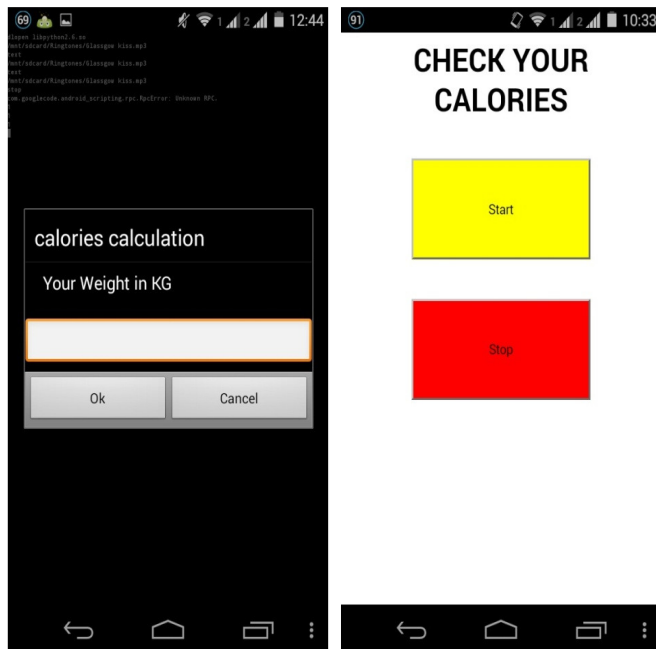


Figure-6
 Calories calculation page

After finish the jog user will see the result on the following html page

Future work: I will work on the following criteria of the application “Juzz Say” in future; those are i. Including more voice enable services. Like: phone calls, Bluetooth applications, Wi-Fi applications etc. ii. The security is not so good. That’s why security building is essential for this application, like: password enabled user account. iii. The application should be more user-friendly. iv. More optimized voice input. Like: noise-proof. v. More attractive User Interface. vi. Including more user languages.

Conclusion

The application has so many things to be developed but, we have seen that the application is very easy to use and we can use the application anywhere. Everything that we need just an internet connection on our android device.

The blind peoples can also use this application.

Now a day, we can see the antisocial activities are increasing day by day. So, if a person needs any help then using the SOS service of the applications he/she can send his/her message with his/her location (latitude and longitude) via SMS to anybody. In this situation I think women have to use the service.

The application is most cost effective than any other applications, because the application uses some heavy weight services but required very less amount of storage. This is the beauty of Python language.

References

1. Python official website: <http://www.python.org/> (2014)
2. <https://code.google.com/p/pybluez/wiki/Documentation> (2013)
3. SL4A official site: <http://code.google.com/p/android-scripting/>(2014)
4. SQLite official Page: <http://www.sqlite.org/> (2014)
5. <http://www.w3schools.com/html/> (2014)
6. Paul Ferrill, Android applications using Python and SL4A, Part 1:Set up your development environment, 19Dec,2010, ibm.com/developer Works/ (2010)