

Smart Alarm

Juncheng Li & Zhicheng Yan

Stevens Institute of Technology
CpE 555: Real-Time and Embedded Systems
Professor Hong Man

1. Background

Android is very popular these days, and it has the largest installed base of all operating systems (OS) of any kind. Android has been the best selling OS on tablets since 2013, and on smartphones it is dominant by any metric. The most important thing is that Android provides a rich application framework that enables people to develop innovative applications and games for mobile devices in the Java language environment. It is easy for developers to exploit it.

In our daily life, it's hard to get up on time. People are becoming less sensitive to the traditional alarm. Some people even fall asleep again after shutting alarm down. So, we design a new smart alarm clock that forces the user to do a simple calculation to refresh themselves to make sure they can get up on time.

2. Abstract

Our project aims to build a smart alarm clock application. Every time it alarms, it will give the user an easy quiz, only when the user's answer by voice is correct, the alarm will stop. The app is based on the Android system, and we develop four modules to realize our purpose. 1) alarm clock module; 2) quiz module; 3) voice to text API; 4) number convert.

Hardware:

Android smartphone: BLU Studio 5.5 res: 480 x 854 pixels size: 5.5 inches

Asus Laptop with Windows 10

Software:

Development IDE: Android Studio

Development Language: JAVA, XML

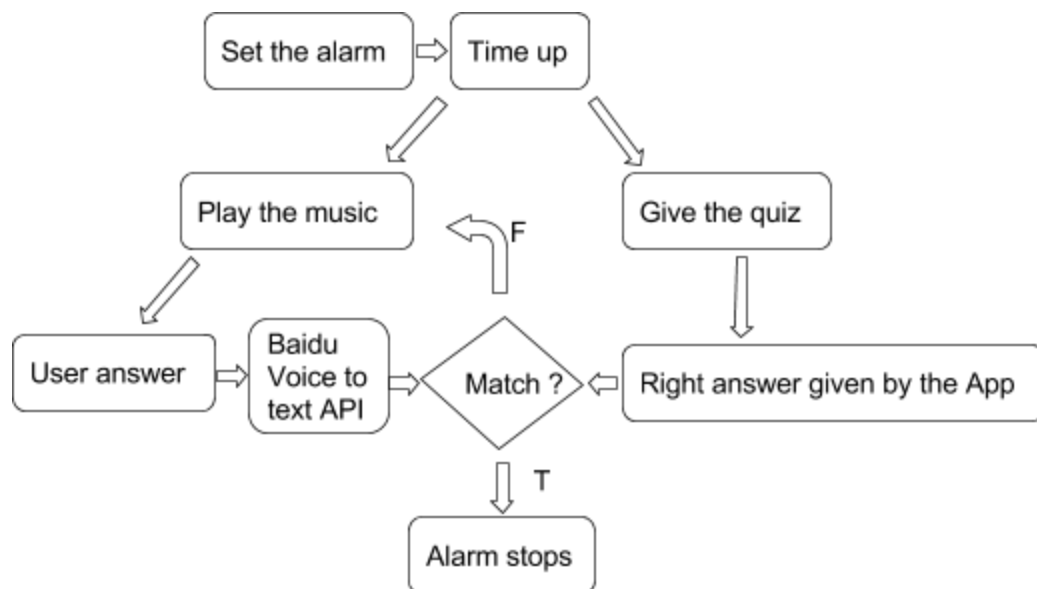
Development Platform: Android 4.2.1

Voice API Supports: Baidu

3. System Design

User sets an alarm. When time is up, the program will initialize a quiz by picking up two random numbers as variable A and B(should be in 1 to 99), then calculates the sum C. Then the speaker will play the ringtone. The quiz will be put in a TextView like "20 + 15 = ? ". When user presses the answer button, the microphone will collect the voice data of the user, submit it to the Cloud, convert the voice data to number D. If number D is equal to number C, the alarm will end. Otherwise, if number D is not equal to number C or the program fails to collect valid voice data, the speaker will keep playing the ringtone.

Flowchart:



4. Modules

Module 1 - Alarm Clock: For this part, we import `android.os.CountDownTimer`. We use mainly 3 functions of this class.

`.start()` - when user set up an alarm, it will activate the countdown timer.

`.onFinish()` - when the timer hits 0, it will generate the quiz. Also it will call `MediaPlayer.start()` to play the ringtone.

`.onTick()` - When the countdown is activated, it will call `TextView.setText()` to show the remaining time.

Module 2 - Quiz: The random quiz is based on `.ranNum()` function. When the time is up, the system will call this function to generate 2 random numbers(a and b), then calculate the sum(c) and wait for the user's answers to be input.

```
int ranNum() {  
    int i = (int) (1 + Math.random() * (50 - 1 + 1));  
    return i;  
}
```

Module 3 - Voice to Text API : The API is provided by BAIDU. First we need to go (<http://yuyin.baidu.com/>) to sign up for API utility. Then download the SDK we need, after put the API library into project library. We are able to:

```
import com.baidu.voicerecognition.android.VoiceRecognitionConfig;  
import com.baidu.voicerecognition.android.ui.BaiduASRDigitalDialog;  
import com.baidu.voicerecognition.android.ui.DialogRecognitionListener;
```

Then we need to set the API_KEY and SECRET_KEY as we got from BAIDU:

```
private String API_KEY = "TbuYqXPGunHjGV6piPyfoMPo";  
private String SECRET_KEY = "b9d6a1023922af2fff4f60fb5090842a";
```

The code we need to use the API has been provided by developer documents:

```

final String recognition_result = "";
Bundle params = new Bundle();
params.putString(BaiduASRDigitalDialog.PARAM_API_KEY, your_api_key);
params.putString(BaiduASRDigitalDialog.PARAM_SECRET_KEY, your_secret_key);
BaiduASRDigitalDialog mDialog = new BaiduASRDigitalDialog(this, params);
mDialog.setDialogRecognitionListener(new DialogRecognitionListener(){
    public void onResults(Bundle arg0) {
        ArrayList<String> rs = results != null ? results
            .getStringArrayList(RESULTS_RECOGNITION) : null;
        if (rs != null && rs.size() > 0) {
            recognition_result = rs.get(0);
        }
    }
});
mDialog.show();

```

And there is one thing we need to pay attention. The API is default to be used for chinese voice to text, we need to set it to english:

```

params.putString(BaiduASRDigitalDialog.PARAM_LANGUAGE,
VoiceRecognitionConfig.LANGUAGE_ENGLISH);

```

Module 4 - Number Convert: We should notice that after the voice data is processed by Baidu API, actually, we don't really get the number. Because of the features of API, we can only get the words of number(i.e. twenty). So we need to convert the words to real number(i.e. 20). We can import java.util.HashMap to solve this problem:

```

public String parse(String str) {
    HashMap<String, Integer> hm = new HashMap<String, Integer>();
    hm.put("zero", 0);
    hm.put("one", 1);
    hm.put("two", 2);
    hm.put("three", 3);
    hm.put("four", 4);
    hm.put("five", 5);
    hm.put("six", 6);
    hm.put("seven", 7);
    hm.put("eight", 8);
}

```

```

hm.put("nine", 9);
hm.put("ten", 10);
hm.put("eleven", 11);
hm.put("twelve", 12);
hm.put("thirteen", 13);
hm.put("fourteen", 14);
hm.put("fifteen", 15);
hm.put("sixteen", 16);
hm.put("seventeen", 17);
hm.put("eighteen", 18);
hm.put("nineteen", 19);
hm.put("twenty", 20);
hm.put("thirty", 30);
hm.put("forty", 40);
hm.put("fifty", 50);
hm.put("sixty", 60);
hm.put("seventy", 70);
hm.put("eighty", 80);
hm.put("ninety", 90);
hm.put("hundred", 100);
hm.put("thousand", 1000);
hm.put("million", 1000000);
int i = 0;
int b = 0;
int c = 0;
String[] k = str.split(" ");
try {
    for (String string : k) {
        if ("hundred".equals(string)) {
            i *= hm.get("hundred");
        } else if ("thousand".equals(string)) {
            b = i;
            b *= hm.get("thousand");
            i = 0;
        } else if ("million".equals(string)) {
            c = i;
            c *= hm.get("million");
            i = 0;
        } else if ("negative".equals(string)) {
            i = 0;
        } else {
            i += hm.get(string);
        }
    }
} catch (Exception e) {
    b = 0;
    c = 0;
}

```

```

}

i += c + b;
for (String string2 : k) {
    if ("negative".equals(string2)) {
        i = -i;
    }
}
String dis = String.valueOf(i);
return dis;
}

```

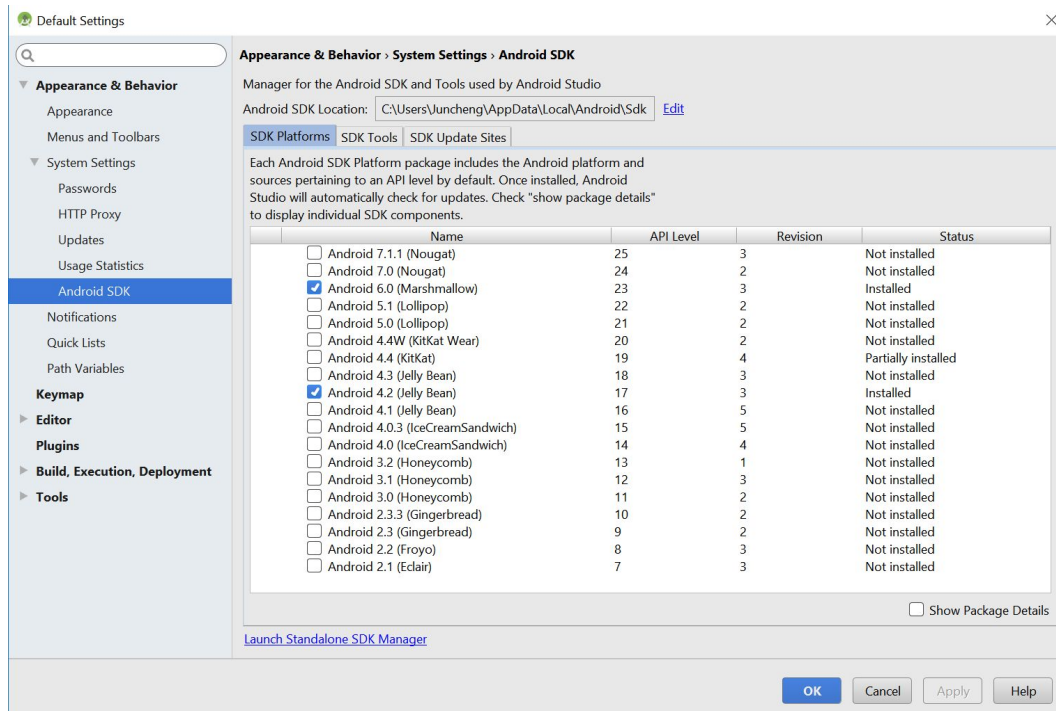
When the system gets a data of words of number from Baidu API, the data will be processed by this .parse() function. This function will split the words by space, then find the real number for each word of number. Finally combine the output to the real number we need.

5. How to Compile/Run

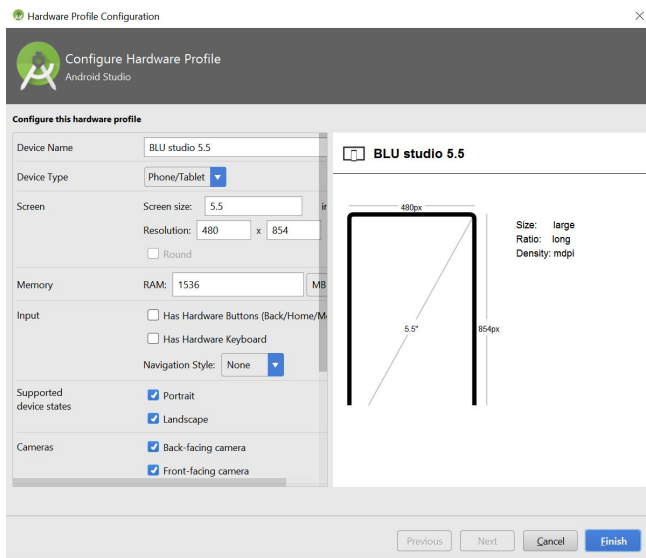
To compile and run the codes, you need to get the Android Studio.

Then click on 'File' - 'New' - 'Import Project...', then choose the project document I upload

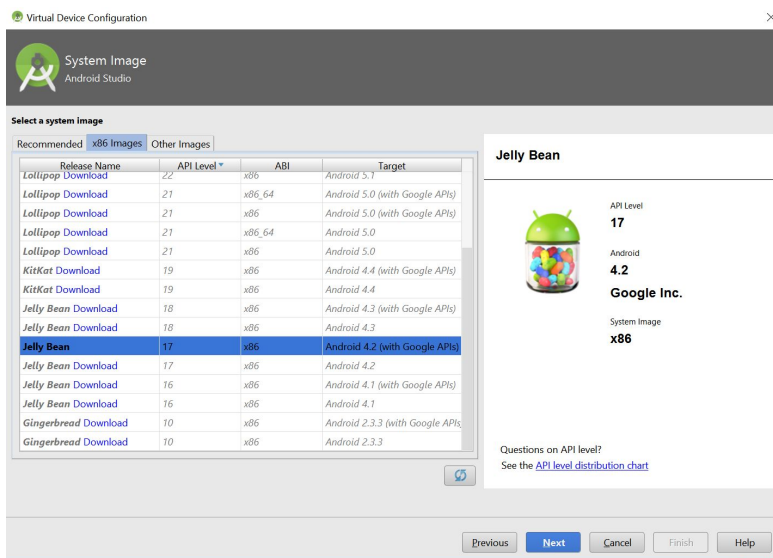
And make sure you have installed the Android 4.2 SDK Platform Level 17(see below)



After that, you need to create a new virtual device, set the display size to 5.5 inch, resolution to 480x854 pixels.



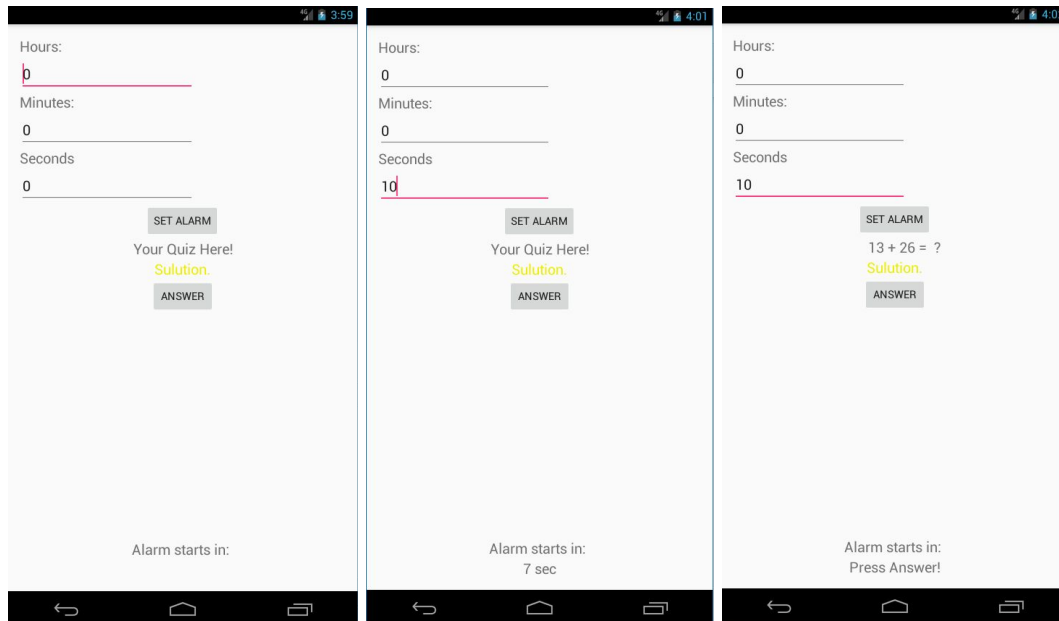
Then click 'Next' and pick 'Jelly Bean' 17 x86 Android 4.2 with Google API'



After those steps, you are all set. You can click on the green triangle and pick the virtual device you make to run.

6. Screenshot of the App

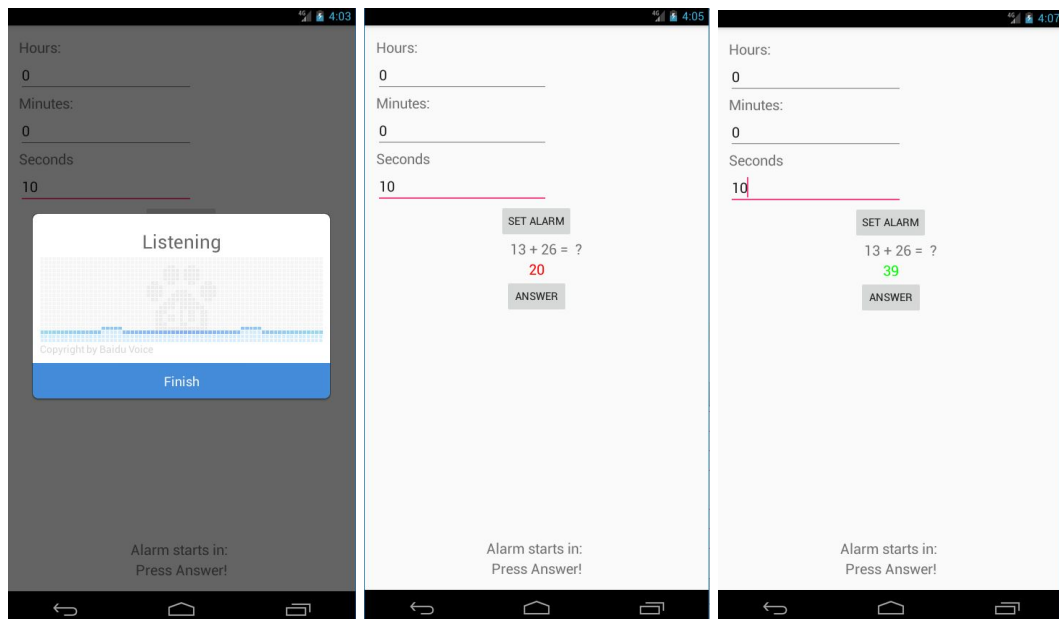
See the screenshots below: (1)open the app (2)start countdown (3)time is up, play ringtone and give out quiz (4)press 'answer' to start voice recognition (5)wrong answer, still play ringtone (6)right answer, stop play ringtone



(1)

(2)

(3)



(4)

(5)

(6)

7. Future Work

To focus on the applicability of the app, we can change the countdown timer into a real clock. To realize this change, we need make the app can grab the time data package of the Android smartphone system and set the app can run in background. Then it will be more powerful in our daily life.

Second, we can add a text to voice API to this app. Every time the user pushes the “Answer” button, we can make the app speaks out the quiz. After that, the user can answer the quiz in the locked screen situation. It will make the app more humanitarian.

Third, the quiz function of this app is easy, we can design a test bank to satisfy different kinds of people. For example, easy addition and subtraction for the children, then level up for the primary school student, then middle school student and etc.

Last thing we can improve is that, we can make a better UI for the app.

8. Reference

1. <http://yuyin.baidu.com/docs/asr/152>
2. <http://blog.csdn.net/u012631267/article/details/19823253>

9. Contribution

Juncheng Li: System Design; Setup Development Environment; Layout Design; Code of CountdownTimer; Code of quiz

Zhicheng Yan: Code of voice to text API Module; Revise of the number convert part; Debug for the application

We have upload the physical hardware testing demo to Youtube, please see:

<https://www.youtube.com/watch?v=NFuYIGPxqDs>

Or you can install \SmartAlarm\app\app-release.apk to your android smartphone to test it!