Multi-Alarm

Yuan LYU
Razvan-Daniel URSACHI
Chenghong XIE

Human Computer Interaction Design

1. Introduction

Multimodal interactive systems are becoming widespread. They now cover many different application domains and support a wide variety of users in the performance of their tasks. It encompasses a wide range of modalities including eye gaze, gestures, touch interaction, two-handed interaction as well as modalities based on multiple inputs on one device. Besides, multimodal interactive system enables users to interact with computers through various input modalities (e.g. speech, gesture, eye gaze) and output channels (e.g. text, graphics, sound, avatars, synthesized speech). This kind of user interface is not only beneficial for enhanced accessibility (visually or motor impaired people), but also for greater convenience (natural input mode recognition) as well as flexibility (adaptation to context of use, to tasks or to users' preferred interaction modalities).

Based on the benefit of multimodal interactive systems, we propose Multi-Alarm which considers about environment and people's activities, which is different from common clock. Basically, people's behaviors will be different during different contexts. For example, when people are running, they may use the alarm to remind them how many time they have run. As our observation, people usually tie their smartphone on their arm. In such situation, when the alarm works, people do not like to use hand to dismiss the clock. Because the alarm is tied in the arm which is inconvenient for them to use hand, also maybe they want to keep running and the hands are not free. Meanwhile, they more like to use voice input to dismiss the clock. Or in that situation, the alarm just rings once which functions as reminding and people do not need to dismiss it.

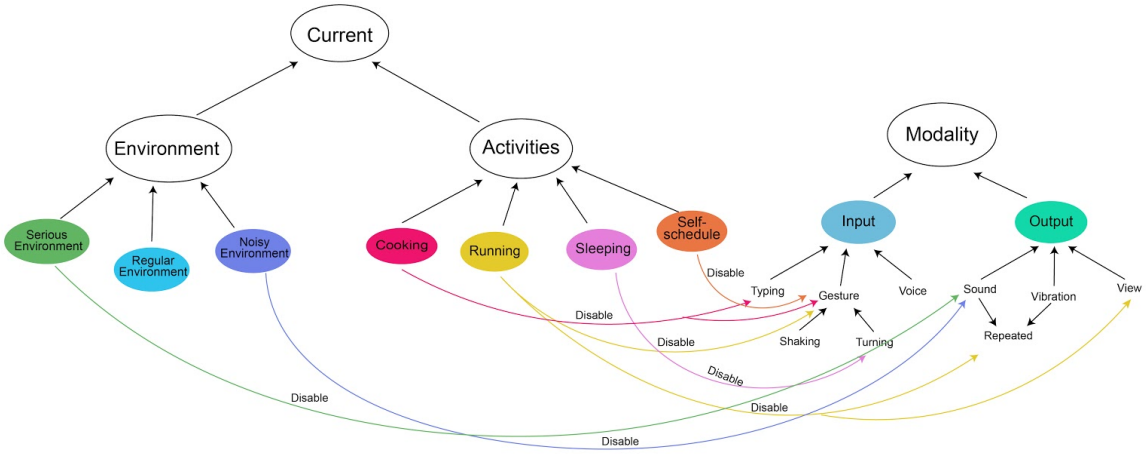
So considering about different behaviors during various context, Multi-Alarm is developed. It not only considers about a variety of human activities, but takes the environment into account, such as serious environment, noisy environment, common environment. Users can choose different activities based on different environments, the system will automatically provide corresponding modalities (input and output).

This report is organised as follows: Section 2 introduces our design concept and user interface. Section 3 presents the implement process, like how it works with some original coding. Section 4 describes the evaluation process base on some scenarios. Section 5 does a conclusion about our work.

2. Design concept

Based on our observations, we conclude some activities and environment people use alarm. And the RDF graph is used to describe our design concept. As the RDF shows, our context is devided into two part: environment and activities. Environment includes serious environment, regular environment and noisy environment. And the activities contain cooking, running self-

schedule and sleeping. As for the modality, there are input modality and output modality. Input modality includes typing, gesture and voice. Output modality are sound, vibration and view. In the input modality, the gesture is devided into shaking and turning. Shaking means people should shake the smartphone strongly to dismiss the alarm which is a difficult way to



dismiss the alarm. While turning means people can dismiss the alarm by turning over the smartphone, which is an easier way to dismiss the alarm. In output modality, sound and vibration has repeated branch. It means the ringing of the alarm can be repeated several times or just work once.
Picture1: RDF graph

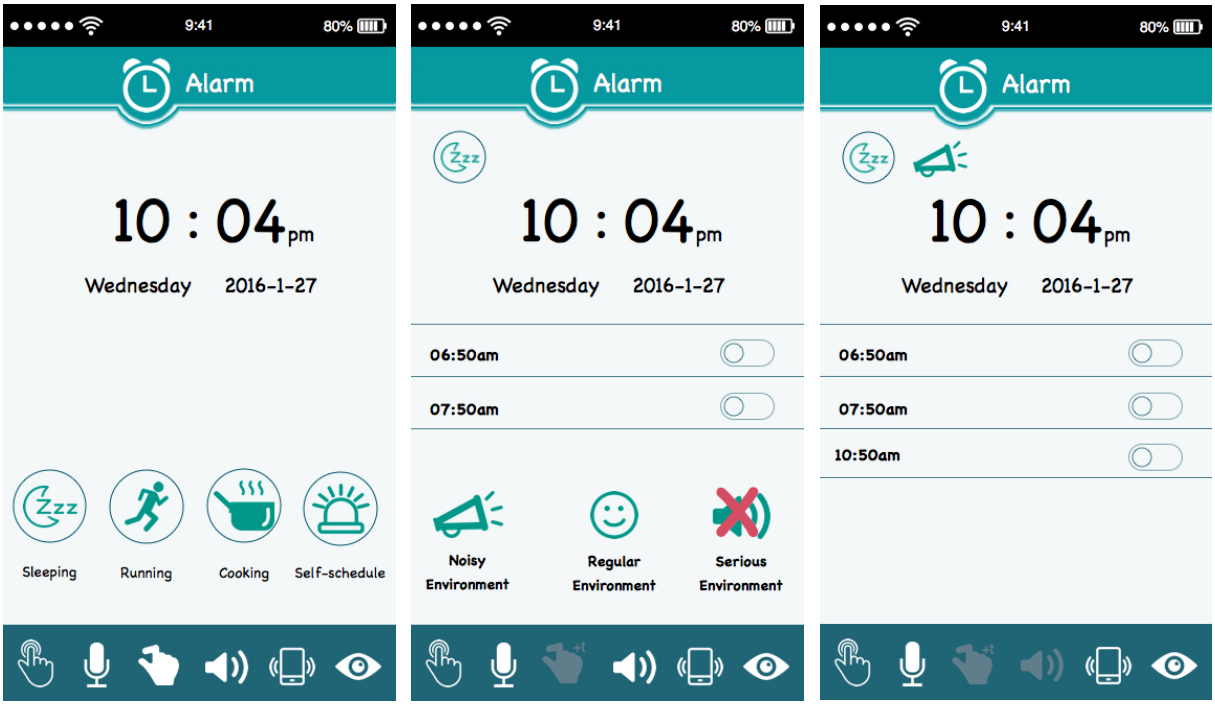
From the view of activities:

1) When people are running, their hands will shake. So we disable the shaking modality which would be useless in that situation. Also, because in normal life, most people will tie their smartphone on their arm, so they do not need to see the interface, the sound or the vibration is enough to remind them the time they have run.
2) When people are cooking, sometimes they should set alarm to remember the time of cooking. But when people are cooking, their hands are not free. They need to do something in the kitchen. Or their hands are dirty, they do not want to use hands to touch the screen. So we disable the typing and gesture modalities.
3) When people are sleeping, they want the alarm to wake them up. But sometimes, sleepy people may turn over the smartphone to dismiss the alarm and continue to sleep. So we disable the turning gesture. It will force people to shake the smartphone strongly to dismiss the phone. In such situation, people would be waked up.
4) When people do the self-schedule activities. It means sometimes people will use alarm to remind them to do something. For example, some people will set a specific alarm which reminds them to take medicine or go to sleep. In order to make sure people look at the information on the screen, we disable the turning gesture. Also it is not necessary for them to shake strongly to dismiss alarm. Normally, people will look at the screen of smartphone when alarm rings and typing the screen to dismiss the alarm. So we disable the shaking gesture.

From the view of environment:

1) In serious environment, like meeting, concert, movie theatre. It should be very quiet. So we disable the sound modality, just use vibration as output modality.
2) In regular environment, nothing happens. We do not disable any modalities.

3) In noisy environment, like train, public place. The sound output is useless. It is not easy for people to hear the sound of alarm. So we disable the sound modality. And strong vibration is used in that situation. Because maybe people put their smartphone in their packet or bag. They can feel the strong vibration when the alarm works.

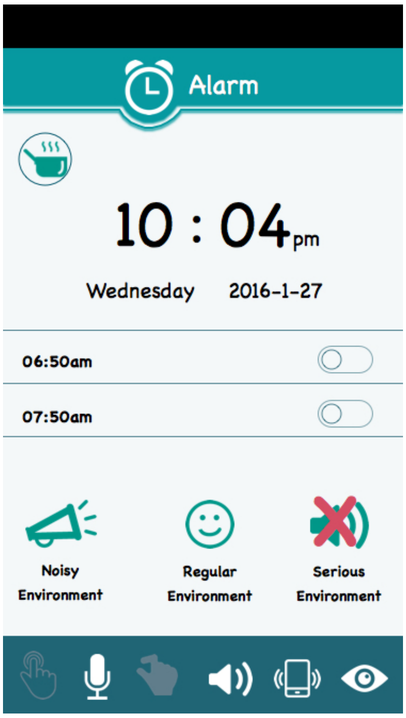


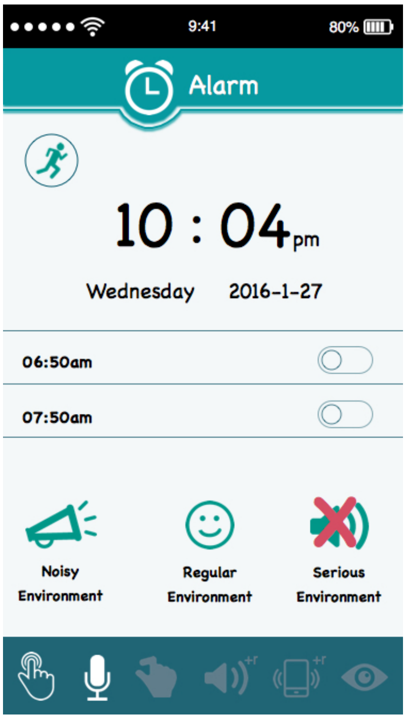
Base on the design concept, we design the multimodal interface of Multi-Alarm (as the pictures below)

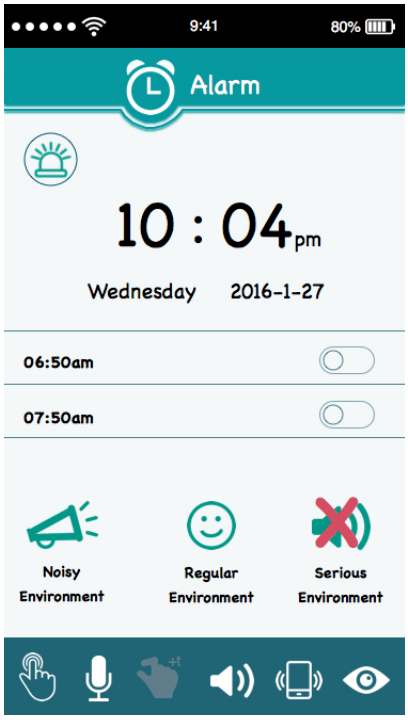
1) In the front page, users can see the time in the up. The four green buttons below time present the 4 activities users can choose. From the left to right is: sleeping, running, cooking, self-schedule. 6 icons in the bottom stand for the 6 modalities (3 input, 3 output). From the left to right is: tying, voice, gesture, sound, vibration, view. When the modality icons are white, it means those modalities are active. If the modality icons turn to gray, it means they are disabled.
2) If users type the sleep button, they can go to the sleep page. In the sleep page, users can see three buttons below time which presents 3 different environments. From the left to right is: noisy environment, regular environment, serious environment. Because in sleep activity, turning gesture is disabled, the gesture icon turns to gray. And in the right up of the gesture icon. There is a mark "+t", which means "turning gesture". So in here turning gesture is disabled.
3) If users type the noisy environment, they will enter the "sleep + noisy" page. In this page, sound modality will be disabled considering about the noisy environment. So in sleep in noisy environment context, the turning gesture and sound modality will be disabled.

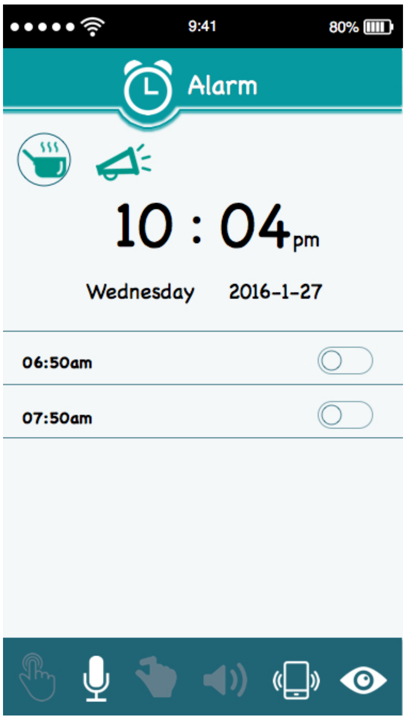
Front page Other interfaces sleep page are as Sleep + noisy below. The principles are almost the same as above. Also, we consider another color

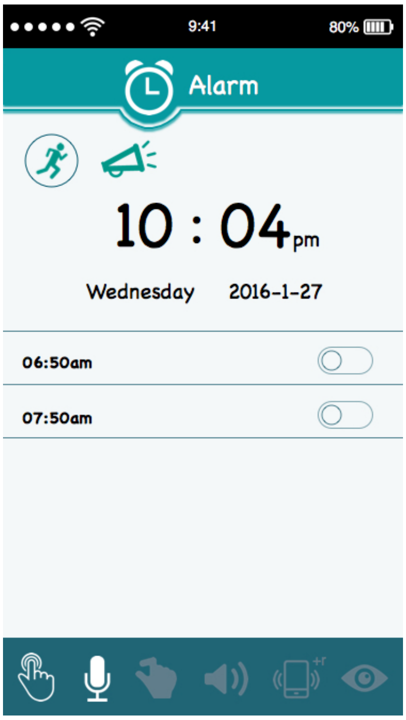
design in the interface which is gray. The logo has two versions too (one is green and the other is gray).

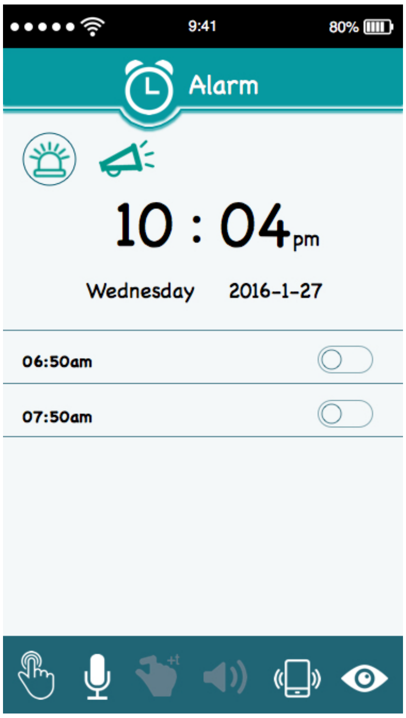

Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am
Noisy Environment
Regular Environment
Serious Environment

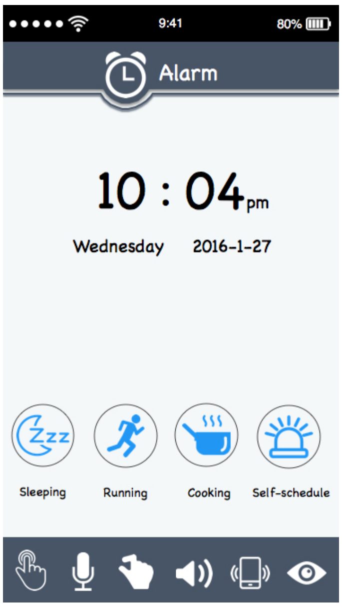

9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am
Noisy Environment
Regular Environment
Serious Environment

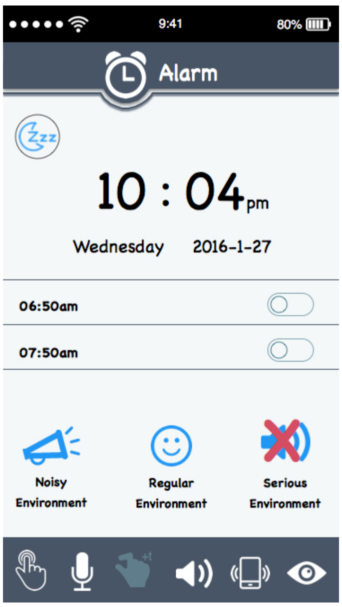

9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am
Noisy Environment
Regular Environment
Serious Environment

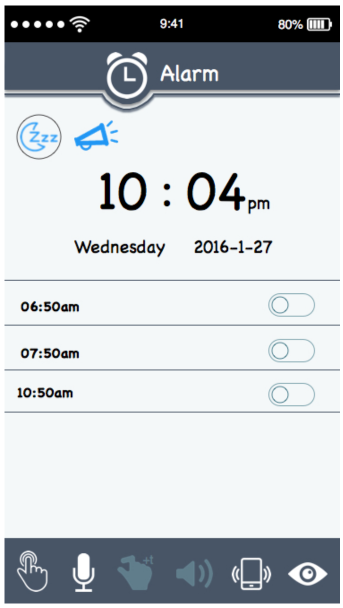

9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am


9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am


9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am


9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
Sleeping
Running
Cooking
Self-schedule


9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am
Noisy Environment
Regular Environment
Serious Environment


9:41
80%
Alarm
10 : 04pm
Wednesday 2016-1-27
06:50am
07:50am
10:50am

3. Implement

RDF part: as the picture shows below, which is the RDF coding.

```
BASE <http://my-uri/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

<current> <has-dimension> <start> .
<current> <has-dimension> <environment> .

<start> rdf:type <activity> .
<start> rdf:value <sleeping> .

<environment> rdf:type <serious> .
<environment> rdf:type <regular> .
<environment> rdf:type <noisy> .

<activity> rdf:type <context> .

<running> rdf:type <activity> .
<cooking> rdf:type <activity> .
<sleeping> rdf:type <activity> .

<modality> rdf:type <context> .

<input-modality> rdfs:subClassOf <modality> .

<gesture> rdfs:subClassOf <input-modality> .
<gesture> <has-property> <disable> .

<shaking> rdf:type <gesture> .
<shaking> <has-property> <disable> .

<turn> rdf:type <gesture> .
<turn> <has-property> <disable> .
```

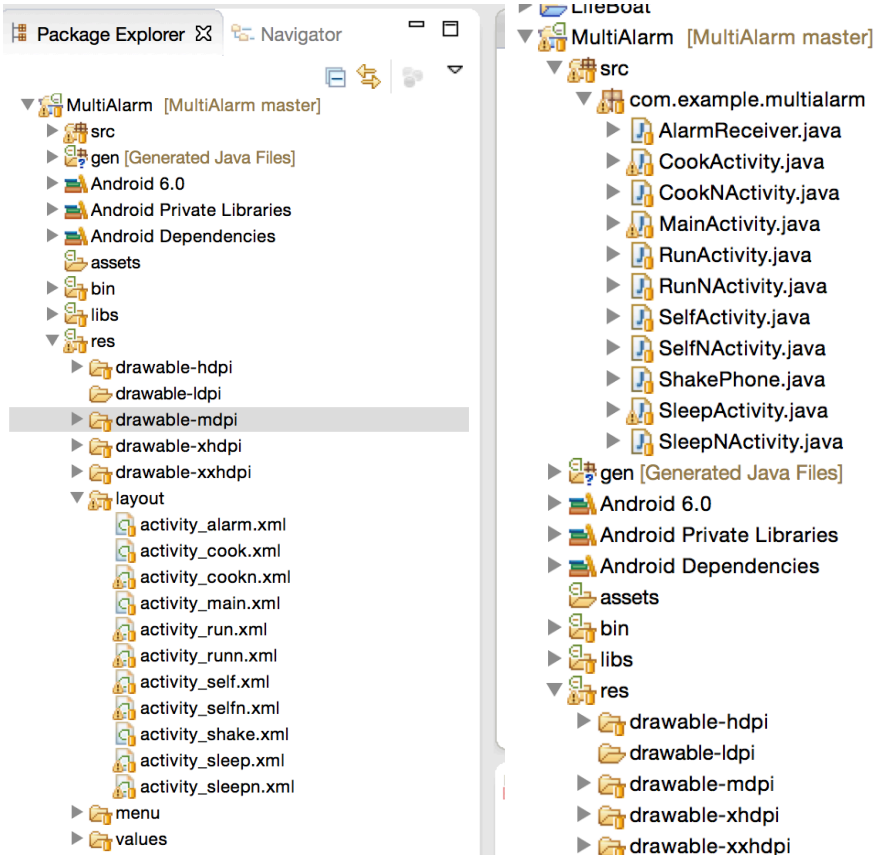
```
<turn> rdf:type <gesture> .
<turn> <has-property> <disable> .

<typing> rdf:type <input-modality> .
<typing> <has-property> <disable> .

<voice> rdf:type <input-modality> .

<output-modality> rdfs:subClassOf <modality> .

<sound> rdf:type <output-modality> .
<sound> <has-property> <repeat> .

<vibration> rdf:type <output-modality> .
<vibration> <has-property> <repeat> .

<view> rdf:type <output-modality> .
<view> <has-property> <disable> .
```

After we get RDF successfully, we write RDF into the java develop environment, but there are some main problems we should solved in Android developing:

1) Using Alarm Manager
   This is basic class for building an alarm application, it can change the status of the alarm, and skip the broadcast
2) Using Broadcast
   Open the activity by broadcast, realize the shake function
3) Using Vibrator and Playing sounds
   Change notification style between vibrate and ring through Media Player and Vibrator class
4) Using Sensor Manager
   In this app, we need Sensor Manager to open the accelerated sensor, then catch the degree of phone shaking to judge the value of the wake up
5) Timer
   Calculate when does the alarm working by using the Timer

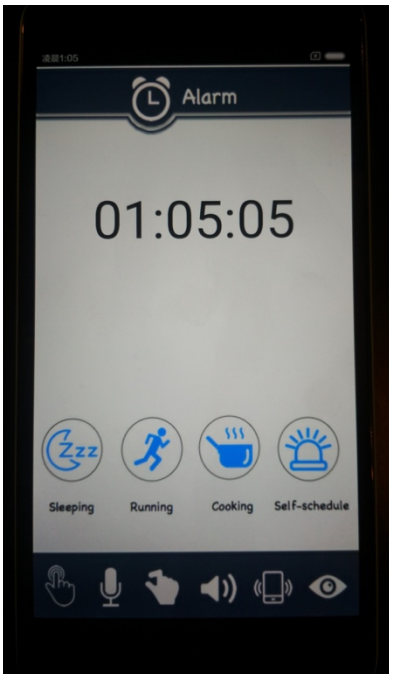
We built the structure for every interfaces, includes alarm for 4 modalities' interfaces, and each modality has 3 different environments. We only implement the noisy environment for our alarm app here. And also, one more interface is for the shake activity and another is for the alarm setting.

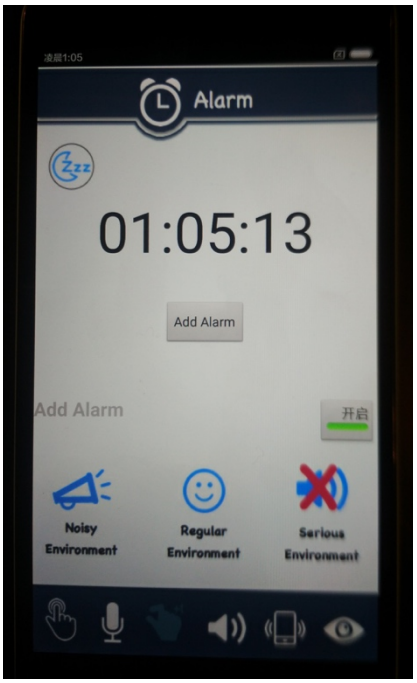
To made app really run in the phone, we still need contacted between interface and java file. All the java files added to one package. AlarmReceiver.java extends the Broadcast Receiver to trigger the alarm. ShakePhone.java is most class to manage the sensors. As for every interface activity, it can call the same alarm but using different values.

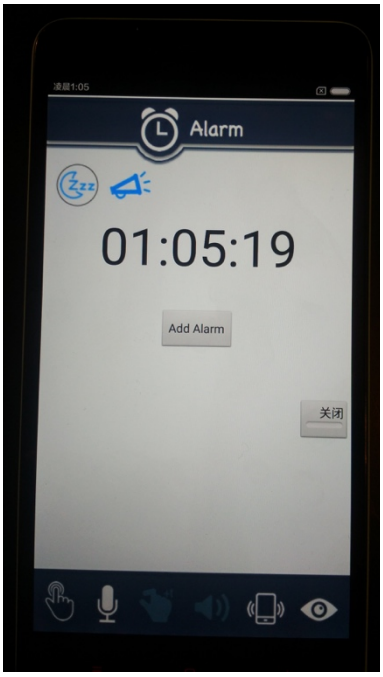


4. Evaluation

After implement, we load the Multi-Alarm into our smartphone and find some users to do user testing. The real interfaces in smartphone are as below:



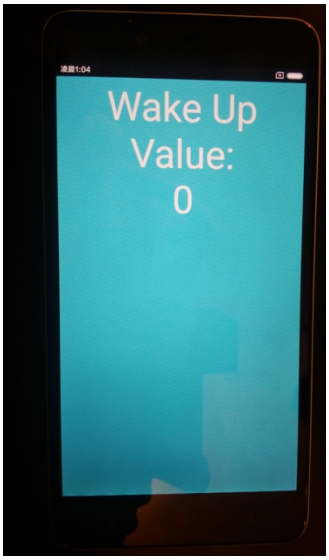


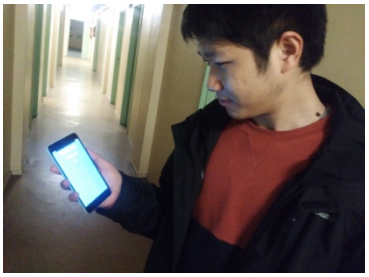


Main page Sleeping page Sleeping + Noisy

We have implemented all the interfaces in design part. At present, the function in sleeping modality is working well. According to this function, we found users to do user testing. Since in sleeping activity, the truing gesture is disabled, users need to shake smartphone strongly to dismiss alarm. When the alarm rings, user can see the message (remind them to shake smartphone to dismiss alarm) in the interface which is as pictures below. Also, the user testing pictures are as below.

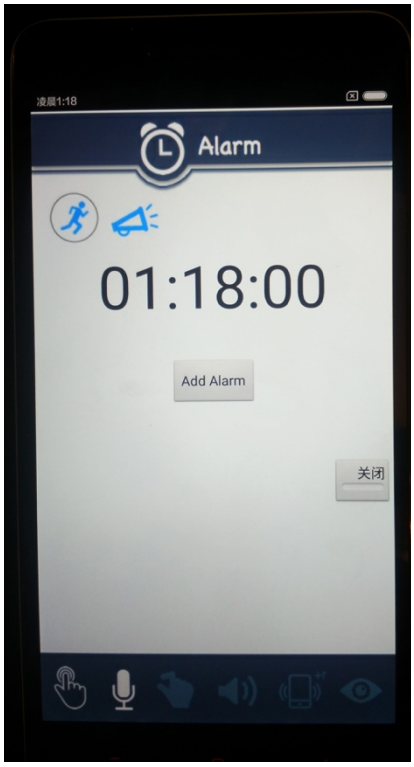
After strong shaking the smartphone, the alarm can be dismissed.

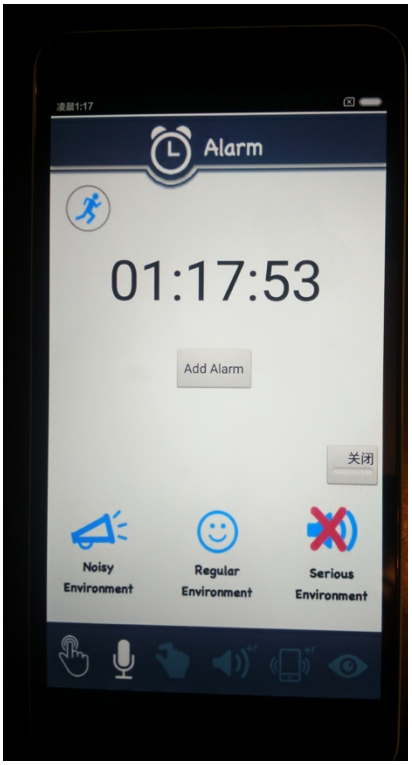


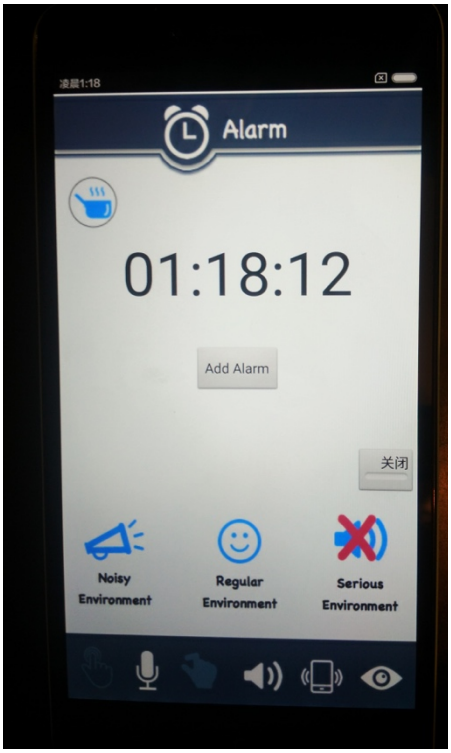


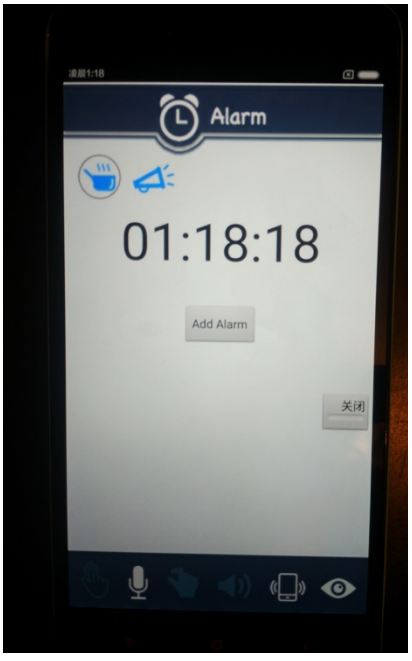
Alarm working page User testing

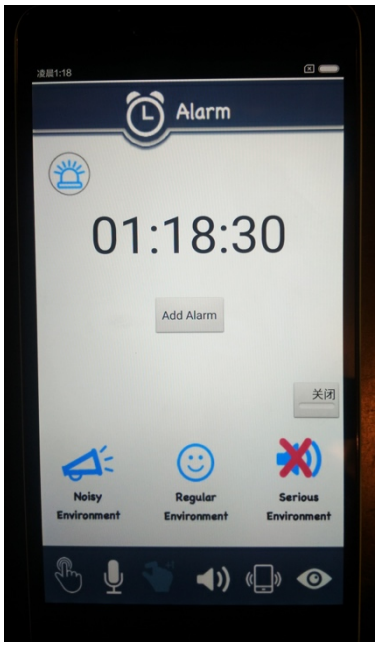
The other interfaces are as below:

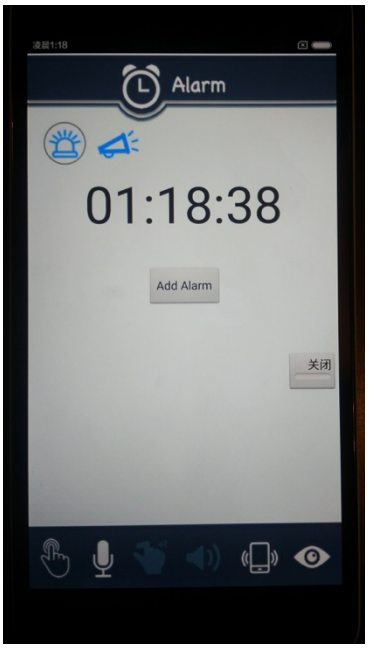












In the user testing, the function in sleeping, and sleeping + noisy environment works well.

5. Conclusion

This report introduces the concept of Multi-Alarm associated with the design process (user behavior observation, interface design, color considering), implementation and evaluation of Multi-Alarm.

Basically, we have realized the alarm function in sleeping modality and noisy environment. And the front-end work is finished, all the interfaces are realized in smartphone. As for functions in other modalities, we will try to realize them in these two days.