Wake-Up Manager Application

1.Introduction & Assumptions

The *wake-up manager application (WMA)* serves as a context-aware personal agent (CPA) that helps users automatically set the wake up alarm in the morning based on users' context. In addition, the WMA application needs to set appropriate ringer mode (vibrate, loud, or loud and vibrate) for the alarm according to the user's surrounding environment. The benefit of having this application is that 1) users do not need to manually set up alarm which saves time, and 2) the WMA can figure out the most proper time for the alarm to keep users healthy.

The following assumptions are made in order for the WMA to work properly:

1) The users store the daily events in the Android calendar so that the WMA can retrieve them.
2) The recommended sleep duration that considered to be healthy is set to be eight hours.
3) The time needed to get to the destination is calculated based on how long it takes to ***drive*** from where user stay for the night.
4) The time needed for user to wake up (including dress-up, having breakfast, etc.) is one hour.

2.Context-aware features

The WMA need to capture users' context in order to set the optimal wake up alarm. The three main context-aware features to determine the time for the alarm are:

1) The time user falls asleep
2) The time of the first events appears in user's calendar
3) The time it takes to reach the destination from where user lives

The WMA need to capture the time users fall asleep but it varies every day so the application need to capture user's context in real time to figure out. In addition, the application need to learn about the time for the event as well as calculate the time needed to get to the destination based on users' current location. The information is dynamic so the application need to adapt and change the time for the alarm in real time.

The WMA also captures the ambient noise around the users when they wake up in order to set the appropriate ringer mode. This user context is dynamic and unpredictable so the application need to retrieve this information in real time and make decision based on the result.

3.Brief implementation detail

The brief implementation detail is explained according to the context-aware feature supported by the proposed application.

To enable the application to capture the time user falls asleep, I decide to use Fence API (https://developers.google.com/awareness/android-api/fence-api-overview.html) to detect if the user current activity is still (not moving). In addition, I use the ambient light and noise sensor to capture environment surrounds user. If the device is not moving within 30 minutes and the environment is quiet and it is dark, then I can conclude that the user is sleeping.

To enable the application to capture events in user's calendar, I decide to use Calendar Provider API (https://developer.android.com/guide/topics/providers/calendar-provider.html) in order to query and retrieve the information including when the event happens as well as where the event takes place.

To enable the application to capture the time needed for driving to the destination, I decide to use Google Distance Matrix API (https://developers.google.com/maps/documentation/distance-matrix/) to retrieve the time to reach destination. Moreover, I need to use google play service (https://developer.android.com/training/location/retrieve-current.html) to get the location users stay for the night.

To enable the application to set the alarm for user to wake up, I decide to use "AlarmManager" class in Android API to set alarm for the user (https://developer.android.com/reference/android/app/AlarmManager.html).

To enable the application to vibrate when the alarm happens, I decide to use "Vibrator" class in Android API to vibrate the phone (https://developer.android.com/reference/android/os/Vibrator.html).

To enable the application to adjust alarm's volume, I decide to use "Audio Manager" class in Android API (https://developer.android.com/reference/android/media/AudioManager.html).

To enable the application to play the ringtone, I decide to user "RingtoneManager" class (https://developer.android.com/reference/android/media/RingtoneManager.html) and "Ringtone" class (https://developer.android.com/reference/android/media/Ringtone.html) in Android API.

To enable the application to use the sensor, I refers to the method used when I developed project 1 (Android + Sensors) for this course.

4.Program Logic

The program should detect the time users fall asleep using Fence API, light sensor and noise sensor. As soon as the time is capture then the application retrieve information from android calendar about the first event (location and start time) occurs in the following morning. Meanwhile the current location for the user is retrieved using google play service. Afterwards, the time required for reaching the destination of the event is calculated.

The following equation is evaluated to determine the time for the alarm:

MINIMUM(***(the time user sleep + recommended sleep hours (8hr))***, ***(event start time - driving time - wake up time (1hr)***)

For example, the application detects that the user sleeps at 11:30 p.m, the first event occurs at 9:00 a.m in the following morning, the time for driving is 45 minutes. Then we can calculate the following:

1) the time user sleep + recommended sleep hours (8hr) = 7:30 am
2) event start time - driving time - wake up time (1hr) = 9:00 a.m - 45mins - 1hr = 7:15 a.m

The minimum of (7:30a.m, 7:15a.m) = 7:15 a.m. Therefore we need to set the alarm time to be 7:15 a.m.

5. Additional Reference for development

1) http://stackoverflow.com/questions/2618182/how-to-play-ringtone-alarm-sound-in-android