Android based walking activity suggestion Application

Min Yang
TU-Darmstadt
Nieder Ramstadter Strasse, Darmstadt, Hesse, Germany
min.yang@stud.tu-darmstadt.de

ABSTRACT

In this paper we present an Android based health care application which is also an implementation of quantified self tool. From the necessary of improving health by prediction medication, exercise as needed, we will elaborate why healthy life can be achieving in such a modern way. Then we head to the technical details about implementing this APP. The problems during this project, the technical stack, the algorithm and a success result will all included. The last paragraph, an emphasis on future work will cover three dimensions to improving this APP.

Author Keywords

Health care; Android APP; quantified self; activity recognition; Google Calendar.

INTRODUCTION AND MOTIVATION

In China, the percentage of elder people predicted to reach 114 million by 2050[1]. A rising requirement of healthy lifestyle to release the burden of medication is on high demanding. We all living in a busy and high pace world. In[2], quantified self is mentioned as the expansion of prediction medicine. How to make the right decision to participate in a timely sports? Basing on quantified self to analysing and giving out the suggestions for sports would be a wonderful solution. This idea is supporting by[3]. So we quantified ourself and we smart enough to know what kind of sports we need to participate in daily life, but sometimes, people find they are lacking the sensing or it just too much concentration needed for them to bring this healthy and scientific way of doing the sporty activities in reality. Applying mobile sensing technologies to recognise human activity is a popular way to their health progress[4] would be a solution.

In this article, an Android based application called GTDNoTi is going to be introduced. During our daily life, we work in front of the computer longer and longer which lead our body in an awkward situation and calling for exercise. Walking could be the easiest way to fits that need on a daily basis. For the purpose of reaching a healthy life style, GTDNoTi meant to help the Android user to monitor their activities and sum up the total walking hours for them in past one week. If the user cannot reach 12 hours of walking in one week. GTDNoTi will consider the user a little fall behind the healthy goal and will try to help by suggesting to create a walking activity in a proper time slot in the calendar of user.

Android can be seen as a landscape changer of mobile market[5] which is picked up as the platform of GTDNoTi. Its economic open source and mature community supporting will lay a solid foundation for the future usability and expansion of GTDNoTi.

In detail, user of GTDNoTi expecting a detection of their own activities automatically, on a daily basis. It detect every five minutes and record the activity and its confidence together into the database, then it checks the cumulative hours of walking activity for last one week daily. If the hours of detected less than 12 in past 7 days, it is not considered to be healthy for the user, then it will find a reasonable time slot for this user to take one hour walking activity by popping up the calendar APP with some preconfigured data. The user can make a much easier decision which is to walk or not basing on their real situation.

The goal of our APP is to play as a helper for user's health care during daily life with respect to the quality of user's choices for living their life. Quantified self from daily activities and analysis basing on the those critical personal data, GTDNoTi can better our user in their life.

The structure of this article will first talk about the what quantification self can better healthy and how tools based on those idea is developing in which stage currently. Two tools published in Google Play Store will be introduced. Then The requirement, running environment, APIs, algorithm and result of GTDNoTi will be present one after another. Finally, the last part will point out the future working directions based on the limitations of GTDNoTi, user's expectation and experience.

RELATED WORK

Kraken.Me[6] is a multi platform that providing tracking tools for mobile, desktop and social. The additional integaration of soft and physical sensors enables fine-grained user profiles. It based quantification tool that tracking user's daily activity and visualising it friendly with grand-truth dataset (valuable data). The main idea behind Kraken.Me is collecting from social connection, for example, Facebook account linked to Kraken.Me account , mobility sensing using Android platform based smart phone, tablet or wearable devices while user on move and digital data. and Kraken.Me mobile[7] is a tracking application based on Kraken.Me framework. It tracks data from sensors in order to retrieve a user friendly profile to managing their time usage for life.

Lables[8] is a quantified self APP for human activity sensing, it can integrated into the Kraken.me infrastructure. To understand human activities corpus. It collect a human activity corpus with ground-truth data. Providing a user interface where users are able to annotate automatically collected sensor data from the mobile, desktop and social with ground-truth, e.g., their performed activitie. It now support Android, later will expend to IOS to cover a wider range of user devices.

Zompono[9] is an application that provide a predicted calendar which is intelligent to ease user's life, its activity recognition using accelerometer data then stored and processed on the cloud service. Integrating cloud and mobile domains is the near future, it aims to consume cloud services from the handset of enriching mobile applications. Mobile phone become the source to understand user in multiple ways[10]. Using accelerometer data and Hadoop MapReduce[11] to ease and accurate the analysis. With the respect of current mobile ubiquitous usage, it fits majority market support. More context-aware services from the cloud that involve the data analysis from other kinds of sensors like magnetic field, in order to find patterns that can be use for predicting actions.

TK Fit can be seeing as a further version based on top of Kraken.Me but focussing on how to make , it is automatically tracking and daily support tool that concentrate on healthy goals of user. Just Kraken.Me, it support Android user. And modifiable for user's activities after monitoring, for example, it monitors user's mood/ activeness and asking for a modification of this user himself/herself. At the top of screen. little add button in mint green asking for adding some activities. What we learned from TK Fit, is that is a complex tool based on the same methodology to reach one's fitness goal by control two dimensions of variable: mood and actual activities. It specialised in the combination or consideration about the human activities with emotions.

There are so many related work that fulfil the activities tracking and quantification in order to maximise the quality of our daily life. Usability, stability and accuracy are the parts that differ a good quantification for healthy purpose tool from one to another.

PROBLEMS AND THEIR OVERCOME STRATEGIES

During the implementation. Some technical problems taking the major time of us to solving it. In this section, I would like to emphasise on some critical parts.

How to recognise the human activities by mobile or wearable devices in a trustable way? Considering the supporting from Google. Activity Recognition AP is come into the picture, this API is stable and capable to recognise human activity based on accelerometer data. By Installing Google Play Service from SDK manager, then with user's permissions 1, with

activity	time	confidence
VARCHAR	DATE	INT
Activity Name	Current Time	The possibility

Table 1. RecordMyActivities

`ACTIVITY_RECOGNITION` we can easily starting to using this API for our goal.

We need to know user's willingness and necessity about activities, thus we using quantified self for analysis. And those data stored in the Android internal database: SQLite[1]. After reading the online tutorial about SQLite usage document, with library android.database.sqlite.SQLiteDatabase, we can access numbers of methods to achieving our goal.

Google Calendar API[2] can enable us to access to Google Calendar APP, but some permissions need to be granted before usage. Like in Figure1, we need to enable `ACTIVITY_RECOGNITIONREAD_CALENDAR` `ACTIVITY_RECOGNITION` and `WRITE_CALENDAR`..

With all the access paths to resources smoothed, we can expand our algorithm. If our application detect that the user less than 12 hours walking happened during last week, then we need searching until one time slot founded. What if this user is really busy, and the schedule is just full. We cannot let this searching endless. Just too much burden on the device and unrealistic. Thus, we design the algorithm only limited to fetch 10 things happened from now and compare them one after another to figure out the right slot for one hour walking suggestion.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mia.gtdnoti">

    <uses-sdk android:minSdkVersion="14" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.MANAGE_ACCOUNTS" />
    <uses-permission android:name="android.permission.USE_CREDENTIALS" />
    <uses-permission android:name="android.permission.READ_CALENDAR" />
    <uses-permission android:name="android.permission.WRITE_CALENDAR" />
    <uses-permission android:name="com.google.android.gms.permission.ACTIVITY_RECOGNITION"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figure 1. enable various permissions

IMPLEMENTATION

GTDNoTi is an Android APP uses: Android SDK 6.0, API level 23. Google Calder API and Activity Recognition API are used as foundation in this APP implementation. You can download the application package from GitHub: **https://github.com/Ashleeeeee/GTDNoTi/blob/master/mobile/mobile-release.apk** I would like to make it an open source APP, thus for developer who want to take a look at the source, you can easily fork it from here: **https://github.com/Ashleeeeee/GTDNoTi**

Architecture overview

GTDNoTi is including a background running broadcast which implemented the activity recognition to monitor user's

[1]http://developer.android.com/training/basics/data-storage/databases.html

[2]http://developer.android.com/guide/topics/providers/calendar-provider.html

activities asynchronously. The data detected go to database for further usage. The main part of GTDNoTi will try to calculate on a daily basis of user's walking activity, calling the Calendar APP if a necessity of walking appointment required. The main architecture looks like Figure2.

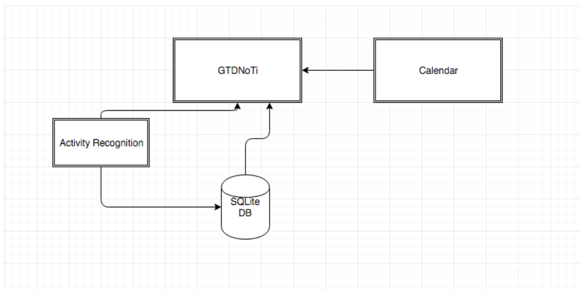

Figure 2. GTDNoTi main architecure

The advantage would be, clear and moduled. Logically it separates the human activity detection logic with analysis, and using the Calendar API for event creation. So the maintenance is scalable.

Interaction with SQLite database

Now let's talk about the implementation details step by step. Starting from the database design and implementation. Since we having s simple request for storing data records, there is only one table called RecordMyActivities like in Table1.

Step one, recognising the activities into database. In ActivityRecognitionIntentService.java, the Activity Recognition API is used to recognise more than 7 activities ("In Vehicle", "On Bicycle", "On Foot", "Walking", "Still", "Tilting", "Running" and "Unknown") for every 5 minutes with its confidence. And a broadcast receiver in onCreate method from MainActivity.java will responsible to record the result into table RecordMyActivities. We use Thread.sleep(milliseconds) to control the detection happening in every 5 minutes. A sample record would looks like Snippet3 in database.

```
/**
 * Add walking event in user's calendar, notify user changes
 */
public void addWalkingEvent() {
    // Calculate total time of walking for past 7 days
    Cursor c = myDatabase.rawQuery("SELECT confidence FROM RecordMyActivities WHERE" +
            " activity ='Walking' and time < datetime('now', '-7 days')", null);
    c.moveToFirst();
    int hours = -1;
    int sumMins = -1;
    int mins = 5;
    for (int i = 0; i < c.getCount(); i++) {
        sumMins = sumMins + mins * c.getInt(0);
        c.moveToNext();
    }
    c.close();
    hours = sumMins / 60;
    // if total waking hours for last week less than 12 hours, tigger startActivity
    // to book an appointment in calendar
    if (hours < 12) {
        refreshResults();
    }
}
```

Figure 3. adding walking event

Algorithm

Step two, calculating the total hours (variable: int hours) on a daily basis for walking activity. In MainActivity.java, method called "addWalkingEvent()" is responsible to figure out past 7 days total hours for walking of our user. Using "sumMins = sumMins + mins * c.getInt(0);" is to balancing the correctness of detection for every record. Just like the snippet from Figure3, if the less than 12 hours, refreshResults() will be called to execute triggering of calendar APP of user. Just like in Snippet 4.

```
protected List<String> getDataFromApi() throws IOException {
    // List the next 10 events from the primary calendar.
    DateTime now = new DateTime(System.currentTimeMillis());
    List<Long> eventDates = new ArrayList<~>();
    List<String> eventStrings = new ArrayList<~>();
    Events events = mService.events().list("primary")
            .setMaxResults(10)
            .setTimeMin(now)
            .setOrderBy("startTime")
            .setSingleEvents(true)
            .execute();

    List<Event> items = events.getItems();

    for (Event event : items) {
        //SimpleDateFormat sdf = new SimpleDateFormat("dd-mm-yyyy");
        DateTime start = event.getStart().getDateTime();
        DateTime end = event.getEnd().getDateTime();

        if (start == null) {
            start = event.getStart().getDate();
        }
        eventDates.add(start.getValue());
        eventDates.add(end.getValue());

        eventStrings.add(
                String.format("%s (%s)", event.getSummary(), start));
    }

    Intent l_intent = new Intent(Intent.ACTION_EDIT);
    for (int i = 0, j = 0; i < eventDates.size(); i = i + 2) {
        j = i + 2;
        if ((eventDates.get(j) - 3600000) >= eventDates.get(i + 1)) {
            l_intent.setType("vnd.android.cursor.item/event");
            l_intent.putExtra("title", "Walk");
            l_intent.putExtra("description", "Hey! It is time to book a walking" +
                    " activity for your healthy goal" +
                    ".");
            l_intent.putExtra("beginTime", eventDates.get(i + 1));
            l_intent.putExtra("endTime", eventDates.get(i + 1) + 3600000);
            //status: 0~ tentative; 1~ confirmed; 2~ canceled
            l_intent.putExtra("eventStatus", 1);
            //0~ default; 1~ confidential; 2~ private; 3~ public
```

Figure 4. trigering calendar APP

On the contrary, if hours bigger than 12, we will set count back to 0 and calculate it again tomorrow. At the mean time, IntentService will keep detecting and recording into the database.

Result

Well, one successful user interaction. The user having a calendar detailed just like Figure5, there are no walking activities more than 12hours during the last week which from 9th of Mar. to 16th of Mar. 16th of Mar. is testing date. This leads to a suggestion for walking activity. The adding interface triggered by GTDNoTi looks like Figure6. Our user feels like adding this activity, then with countable inputs, a changed calendar results in Figure7.

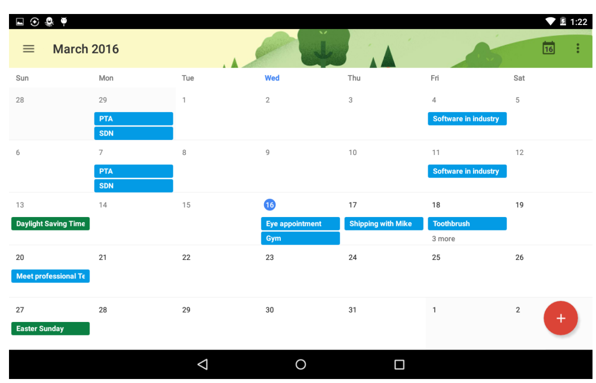

Figure 5. User calendar details

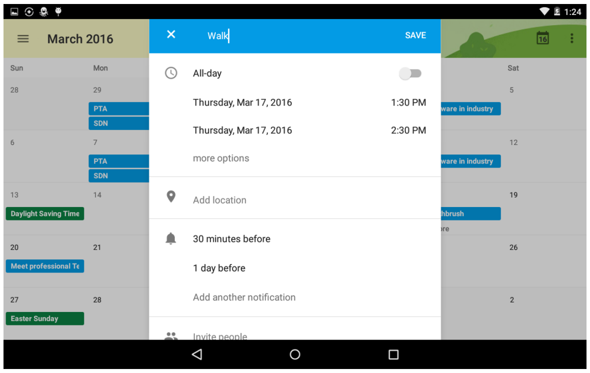

Figure 6. Interface for adding walking event

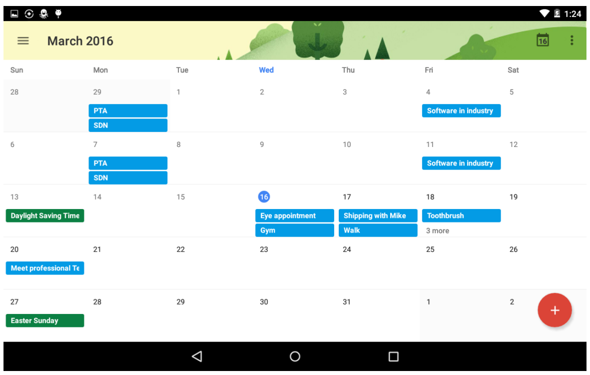

Figure 7. User calendar with added walking event

CONCLUSION AND FUTURE WORKS

GTDNoTi can be a helper during our healthy life style pursuing for regular user. Simple but direct way to reach your goal from digital helper. Base on this purpose, Three dimensions should considered for improving in the future.

First would be the logics which more activities suggestions should be provided, for example, if user walking less than 12 hours but running more than 10 hours during past week, then it is totally fine for him/ her to skip for exercise.

Second would be getting more fun by focusing on human computer interaction, for example, if the user add walking activity, then APP can praising user. To combine those two together, the UX can increasing sky high.

The last but not least would be security, since it based on collecting user's daily activities, those data is quit personal which should under protection rather than leak out improperly.

ACKNOWLEDGEMENT

This work has been helped by Telecooperation research group from institute of computer science, Technical University of Darmstadt. Specially supporting from Mr. Christian Meurisch, Mr. Benedikt Schmidt and Mr. Martin Schmitz.

REFERENCES

1. Holmes, Alex. Ageing in China: health and social consequences and responses. In International Journal of Epidemiology, IEA, (2012), 772-775.
2. Swan, Melanie. Health 2050: The realization of personalized medicine through crowdsourcing, the quantified self, and the participatory biocitizen. In Journal of Personalized Medicine, published by Molecular Diversity Preservation International IEA, (2012), 93-118.
3. Swan, Melanie. The quantified self: Fundamental disruption in big data science and biological discovery. In Big Data, published by Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, (2013), 85-99.
4. Kunze, Kai and Iwamura, Mikio and Kise, Kenji and Uchida, Seiichi and Omachi, Shinichiro. Activity Recognition for the Mind: Toward a Cognitive" Quantified Self". In Computer, published by IEEE, (2013), 105-108.
5. Margaret Butler. Android: Changing the Mobile Landscape. (2011).
6. Srirama, Satish Narayana and Flores, Huber and Paniagua, Carlos. Kraken. me: multi-device user tracking suite Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, organised by ACM, (2014), 853-862.
7. Schweizer, Immanuel and Bartl, Roman and Schmidt, Benedikt and Kaup, Fabian and Muhlhauser, Max. Kraken. me mobile: the energy footprint of mobile tracking In Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, organised by IEEE, (2014), 82-89.
8. Meurisch, Christian, et al. Labels: quantified self app for human activity sensing. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, oragnised by ACM,(2015), 1413-1422.
9. Srirama, Satish Narayana and Flores, Huber and Paniagua, Carlos. Zompopo: Mobile calendar prediction based on human activities recognition using the accelerometer and cloud services. Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011 5th International Conference on, organised by IEEE, (2011), 63-69.
10. Ravi, Nishkam and Dandekar, Nikhil and Mysore, Preetham and Littman, Michael L Activity recognition from accelerometer data. In AAAI,(2005), 1541-1546.
11. Woo, Jean and Kwok, T and Sze, FKH and Yuan, HJ AHadoop in practice. Published by Manning Publications Co., (2012).