

Location Based Mobile Apps Development on Android Platform

Bhuvana Sekar and Jiang B.Liu
Computer Science & Information Systems Department
Bradley University
Peoria, Illinois, U.S.A.
jiangbo@bradley.edu

Abstract. This paper describes a TriTHEApp Android application, designed to serve as a location based time saving and safety ensuring application. In other words TriTHEApp is a location based service, with three features, 1) Tool to announce arrival, 2) Handset theft monitoring and 3) Emergency call feature integrated in it. Tool to announce arrival service is designed to make client's uncertain waiting time known, handset theft monitoring is designed to track user's android device, and emergency call feature is designed to ensure personal safety of the android device user. Development of this application led us to identify few usability concerns and to propose solutions to solve these concerns.

Keywords: Mobile computing, Mobile App development, Location based service.

I. INTRODUCTION

Currently, there are more than 1.5 million android devices [1] are activated in each day and more than 50 billion app downloads from Google play [2] in each month. This indicates the widespread acceptance and usage of mobile device such as android devices among users. There are more and more mobile applications developed on Android operating system. This is because android is the main environment that combines the following features [3]:

- A truly open, free development platform based on linux and open-source.

- A component-based architecture inspired by internet mashups.
- Tons of built-in services out of box.
- Automatic management of the application lifecycle.
- High quality graphics and sound.
- And finally, portability across a wide range of current and future hardware.

Android applications are developed in Java language using the android Software Development Kit (SDK). The development process of android applications can be divided into four phases [3] as illustrated in the following figure.

With tremendous growth in development and utilization of mobile (android) applications, the corresponding impact is also felt on development of new application softwares for personal computers. Following are some of the reasons for why the mobile applications are more preferred over desktop applications in the present days [4].

- Mobile applications give its users flexibility and convenience. As the smart-phones and android devices are always switched on, users can easily access information and services through various applications.
- Mobile devices can provide context sensitive information which the desktop counterpart may not be able to provide.

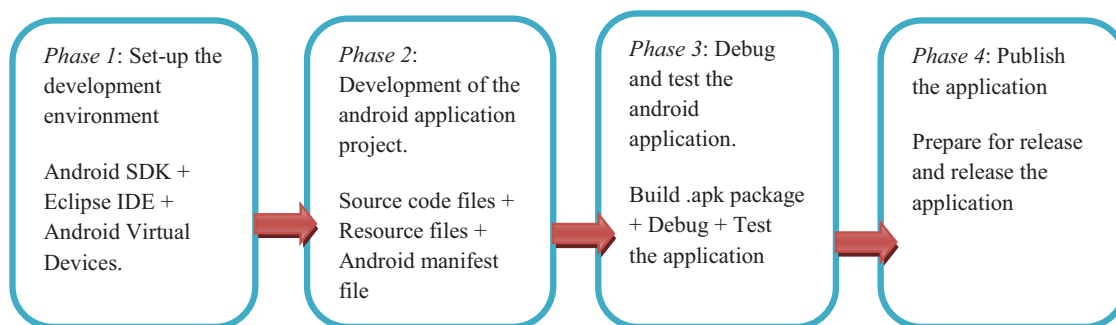


Figure 1. Phases of Android Application development

Location based apps is a good example for this case.

- Business owners can better reach their customers through mobile applications. Big businesses like Amazon, eBay, Target, Gap and many others, started promoting their brands and merchandise with mobile apps a few years ago. Smaller businesses also are quickly following up the suit.
- Mobile applications are easily distributed to its users through app store (for android, Google play is the official app store).
- Mobile applications are found to promote increased user-engagement.

In this paper, we will describe the development of an Android application and to address usability issues in mobile app development to identify the usability concerns and to propose solutions to solve these problems.

TriTHEApp Android Application

In daily life, almost every one of us would have faced with at least one of the following situations depicted in the figure 2.



Figure 2. A few concerns in everyday life

As most of us own a mobile device, the solution to these problems can be provided with Tri App:

To be on time

Help to find your phone

Emergency call

This android application is an integration of three features: 1) Tool to announce arrival, 2) Handset theft monitoring, and 3) Emergency call. The tools used for developing this application are Eclipse IDE with ADT (Android Developer Tools) plug-in and Android SDK (Software Development Kit). This application prototype is developed at API level 16 (Android Jellybean 4.1). The reason behind choosing this API level is that Android Jellybean OS release has established itself as the dominant version of Android that accounting for 40% market-share [5] on android devices and it offers a host

of feature improvements. Design of each feature of the application is described in detail in the following sections.

II. TOOL TO ANNOUNCE ARRIVAL FEATURE DESIGN

Waiting is frustrating, demoralizing, agonizing, aggravating, annoying, time consuming and incredibly expensive” [6]. In the article ‘The Psychology of Waiting Lines – by David Maister’, it is stated that “Uncertain Waits are longer than Known, Finite Waits”. Clients who arrive early for an appointment will sit contentedly until the scheduled time, even if it is a significant amount of time in an absolute sense. However, once the appointment time is passed, even a short wait, say five minutes grows increasingly annoying, if they are uncertain about how long they have to wait to receive the service and thus leads to unsatisfied customers. To address this problem, the TriTHEApp application we developed includes a ‘Tool to announce arrival’ feature, which is a modified implementation of GPS tracing to aid the clients to know their absolute wait time/wait time closer to absolute, by tracking the whereabouts of the service provider (person with whom the appointment has been made) and calculating the time required for them to reach the meeting location based on their current location.

2.1 Feature Specific Components

The main components of the Tool to announce arrival feature is depicted in the figure 3.

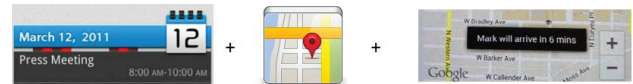


Figure 3. Components of tool to announce arrival.

It includes,

- A scheduler to make note of the appointment details – date, time and any comment about the meeting.
- Google maps API and GPS receiver to trace the location.

2.2 Working of the feature

- Both the client and the service provider should have the application installed in their android devices.
- The service provider is required to add the client’s contact (the person with whom he/she wishes to share his location), location of the meeting and appointment date and time.
- The client is also required to complete the above step.
- At only five minutes (this time interval is configurable by the user with the constraint that selected time should be before the appointment time, and this time interval constraint for GPS tracking is added to address the

privacy issues of the user and also smaller the time interval lesser will be the power drain of the device) before the appointment the users (client and the service provider) can track the location of each other with Google maps and an estimate of the time required by the other person to reach the meeting location is provided to each.

2.3 Auto-start of this feature as a service

A location based application requires GPS (Global Positioning System) of the device to be turned on and the application using GPS tracing to be started to locate the device. There may be situations where an irresponsible service provider may forget to start the application or voluntarily avoid starting the application to share his location with the client, when traveling to fulfill the appointment, in which case the application is rendered useless.

To address this usability concern this feature is required to be auto-started as a service at the time pre-set by the user before the appointment time, so that GPS tracing can be done to provide client the required information without requiring the service provider to start the application manually. An SMS “appointment on wait” is sent to the service provider at the fixed time, which triggers the GPS tracing to start in the service provider’s android device to provide client the required information. Tool to announce arrival is bidirectional, hence the above sequence is repeated in the client’s android device to provide service provider the same information about the client.

Following is the code snippet for auto-start of an application upon receiving a specific SMS that acts as the trigger.

```
/* AndroidManifest.xml including permissions to receive sms
*/
<receiver android:enabled="true"
android:name=".AutostartReceiver"

android:permission="android.permission.RECEIVE_SMS">
    <intent-filter>
        <action
android:name="android.provider.Telephony.SMS_RECEIVE
D" />
        <category
android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>
[.]
<uses-permission
android:name="android.permission.RECEIVE_SMS" />
<uses-permission
android:name="android.permission.READ_SMS"/>
```

[.]

```
public class AutostartReceiver extends BroadcastReceiver{
    public void onReceive(Context context, Intent intent) {
        // Service triggered to autostart to perform GPS
        tracing
        Intent i = new Intent(context,
        AnnounceArrivalBackground.class);

        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(i);
    }
}
```

III. HANDSET THEFT MONITORING FEATURE DESIGN

Losing an android device can be a very unfortunate situation to its user, since mobile device is expensive and carry so many functionalities and user’s private information. There are many applications available in the android market to locate a lost/stolen android device. Since TriTHEApp is designed to be a location-based safety time saver application, handset theft monitoring feature is included in it as one of the main features of the application. It includes logging feature to log android device’s location at set time intervals, hence when the lost phone is switched off, its last location detail logged can be useful to find the device.

3.1 Feature Specific Components

The main components of handset theft monitoring feature are depicted in the following figure 4.

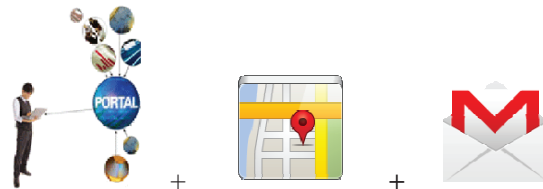


Figure 4. Components of handset theft monitoring.

It includes

- Application’s internet portal where the user has to create an account registering his device, which will then aid to locate the lost device.
- Google maps API with GPS service to locate the device.
- Mail service to log the devices location details in the mail address provided by the user.

3.2 Working of this feature

- User is required to register his/her account with the application's internet portal.
- User can then choose the mode of logging details, i.e., to send details to the application's internet portal, mail to an email address or FTP to a server.
- The feature will be started on the click of the 'log' button, and the GPS will be activated on the android device.
- The GPS tracing begins as a background service and tracks the device's location.
- The logging frequency can be configured by the user that is time interval at which the feature is required to send the tracking details (location and time) can be defined by the user. If the android device is lost and switched off, this configuration helps to determine at least the most recent location of the lost device which can aid in locating it.

IV. EMERGENCY CALL FEATURE DESIGN

This is the third main feature of TriTHEapp, designed to ensure personal safety of the android device owner. An individual's personal safety is always at question when walking alone at night or when going on a lone adventure trip. When caught in an unfortunate situation everyone wishes there is a help to find them. To make it possible for help to find the person in need, Emergency call feature has been designed to be one of the features in the TriTHEapp application to turn the user's android device into an ultimate safety device.

4.1 Feature Specific Components

The main components of the emergency call feature are depicted in the following figure 5.

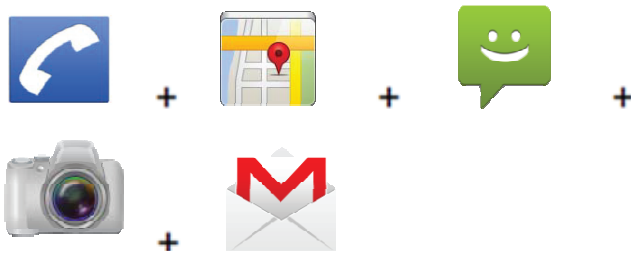


Figure 5. Components of Emergency call feature.

It includes,

- Phone call service to make the call to the emergency contacts.
- Google maps API with GPS service to track the location.
- Short messaging service to message the victim's (application user's) location to the emergency contacts.
- Camera to capture the scene, which can serve as an important evidence.
- Email service to mail the compressed video file to the mail address provided.

4.2 Working of the feature

- User has to choose the emergency call feature from the main menu of the application.
- During the first usage, the user is required to select the contacts to be added to the emergency contact list, ordered based on priority (this list can be edited anytime).
- The user is required to provide an email address to which the video captured has to be compressed and sent.
- Whenever walking alone in unsafe locations, the user can open this feature and push on the toggle button and hold on to it.
- If the button is released without being pushed off, the following events start in a sequence.
- Automated emergency call is made to the first contact in the emergency contact list.
- The current location of the user/victim is text messaged to all the contacts in the emergency contact list.
- The camera is switched on to video record the scene for a pre-set time interval.
- The video is then saved, compressed and mailed to the email address already provided.
- If the user is on an adventure trip and is unable to hold onto the push button, he/she can set a timer within which he can check back to the application, else his location will be text messaged to his emergency contacts notifying that help is required.

4.3 Battery power reserve for Emergency call feature

Many research have been carried out to investigate how to minimize the power consumption of the android applications and many power saving apps are available in Google Play apps store [2]. These apps aim at controlling different android device's features like Wi-Fi, 3G and 4G connections, brightness level, CPU frequency, GPS and more to prolong battery life.

Another related research is context aware battery management for mobile devices [7], it is based on three principles:

- The availability of crucial applications to users should not be compromised by non-crucial applications.
- The opportunities for charging should be predicted to allow devices to determine if they have scarce or plentiful energy, instead of using absolute battery level as the guide.
- Context, such as location information can be used to predict charging opportunities.

Context aware battery management is related to our android application in the sense that, even when an android powered device has an Emergency call feature installed in it, it is rendered useless when the battery power runs out at the needed situation. To overcome this problem a battery reserve component can be added to the Android operating system that monitors the battery life. when the battery falls below a threshold it stops the processes/applications that are not in the crucial applications list of the component and those of which are currently running in the device, either by committing to the current state or by killing it and preventing them from being started again until the device is connected to an external power supply. In this way only the phone call service, the emergency call application and messaging service (applications in the crucial list) be available for usage during limited battery life, thereby being helpful in emergency situations.

The following figure 6 depicts the flow of activities of the battery reserve component.

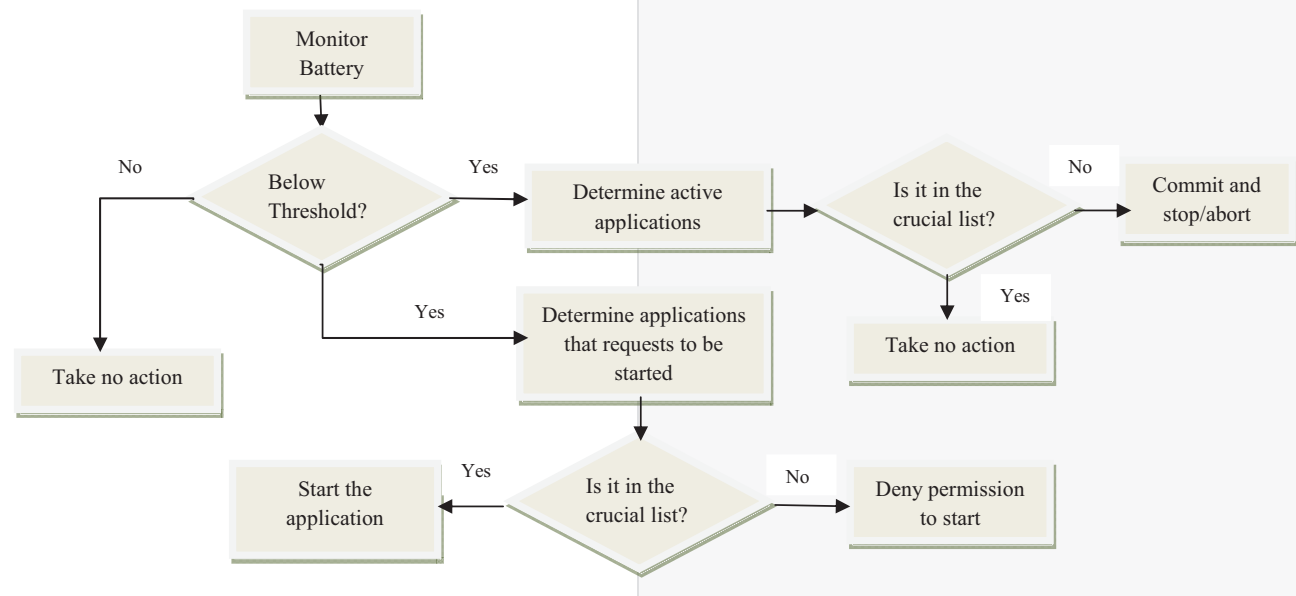


Figure 6. Flow of activities of the battery reserve component

Following is the Code snippet to monitor the battery status of an android device:

```

/*AndroidManifest.xml file including permissions to monitor
the battery status*/
<receiver android:enabled="true"
android:name=".MonitorBattery"

```

```

android:permission="android.permission.BATTERY_STATS"
>
    <intent-filter>
        <action
android:name="android.intent.action.BATTERY_CHANGED
" />
        <action
android:name="android.intent.action.BATTERY_LOW" />
    </intent-filter>
</receiver>
[.]
<uses-permission
android:name="android.permission.RECEIVE_SMS" />
[.]

public class MonitorBattery extends BroadcastReceiver{
@Override

```

```

public void onReceive (Context context, Intent intent)
{
    //Determine the current battery level
    IntentFilter ifilter = new
    IntentFilter(Intent.ACTION_BATTERY_CHANGED);
    Intent batteryStatus =
    context.registerReceiver(null, ifilter);

```

```

        int level =
batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -
1);
        int scale =
batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -
1);
        float batteryLife = (level/(float)scale)*100;
        if(batteryLife < threshold)
        {
            // Stop and deny permission for start of
applications not in the crucial list
        }
    }
}

```

V. CONCLUSION

With the rapid proliferation of mobile devices with android operating system, android applications will continue to create a buzz in our lives. It is transforming businesses. Industries as enterprises are using mobile applications to give customers, employees and suppliers access to data when and where they need. TriTHEApp android application that we developed will serve as a very useful application to its users upon release to the market. It will enhance the customer service of the android devices and device owner's personal safety. The development

process presented in this paper identified some usability issues and the solutions proposed to solve them. Two importance features are auto-start of the applications at required times to make the service available and smart battery life management to reserve limited battery life at times when the android device is running out of power. These features among many others will make an android smart devices smarter companion to its owners.

REFERENCES

- [1] Google activating 1.5m Android devices per day, keeping steady growth in 2013 – Pocket-lint, 17 April 2013, from <http://www.pocket-lint.com/news/>.
- [2] Android apps to overtake Apple – The Telegraph, 03 Jun 2013, from <http://www.telegraph.co.uk/technology/news>
- [3] Application development workflow from <http://developer.android.com/tools/workflow/index.html>
- [4] Impact of Mobile Apps on PC Software - My portable world, 19th July 2012, from <http://myportableworld.com/posts/impact-of-mobile-apps-on-pc-software/>
- [5] Jelly Bean tops 40% market share on Android devices - TopNewsToday.org, 03 Aug 2013, <http://technology.topnewstoday.org/technology/article/7135764/>
- [6] David H. Maister, (2005), Columbia University, The Psychology of waiting lines http://www.columbia.edu/~ww2040/4615S13/Psychology_of_Waiting_Lines.pdf
- [7] Nishkam Ravi, James Scott, Lu Han and Liviu Iftod, IEEE conference, Hong Kong, 17-21 March 2008, Context aware battery management for mobile phones, retrieved from <http://research.microsoft.com/pubs/76681/cabman-percom08.pdf>.