Developing an Android Application

Evan Clements  
nave88@gmail.com  
GSET

Zachary Klapwald  
zdklapwald@comcast.net  
GSET

Revan Sopher  
rsopher@gmail.com  
GSET

NJ Governor’s School of Engineering & Technology 2011

1 Abstract

Driving under the influence is a common and serious problem in today’s society. In 2009 alone, over 10,000 people were killed in alcohol related car accidents in the United States [1]. We decided, therefore, to make an application that would encourage safe drinking and reduce the number of drunk drivers on the road. Named ”Bye Bye DUI”, our app is activated when the user reaches a bar or party, and stores the address and coordinates of their location using the phone’s Global Positioning System. If the user gets a previously specified distance away from their initial location, an alarm will ring and the user will be prompted to take a test that measures sobriety through balance and coordination. If the user fails the test, the phone will automatically send a text message to a designated contact with the user’s location and inform them that they are not fit to drive, so that the designated contact can come help the user out.

2 Introduction

Drunk driving is a problem in the United States despite widespread efforts to educate people about the effects of alcohol. This is in part because one of the effects of drinking alcohol is the loss of judgment. A relatively small amount of alcohol can influence a person’s judgment enough that they might make a poor decision that they would recognize as wrong when sober [2], such as driving drunk. Even a person who is fully aware of the effects of alcohol can make dangerous decisions while under its effects.

Our application is meant to help people like this and prevent them from making such a mistake while intoxicated. A user arriving at a bar or party can activate the application through a touch of a button, selecting the size of the area, having already designated preferences such as a contact and the message to send to that contact if the user fails the sobriety test. The app then stores the user’s GPS coordinates and continues to run as a background process, tracking the user’s location using the phone’s built in GPS. If the user exits the previously designated area, the phone will ring an alarm and prompt the user to take a sobriety test. The test has the user holding their phone out and attempting to keep the screen facing straight up for twenty seconds. This is a test of motor skills, balance, and coordination, all traits impaired by alcohol consumption. If the test is failed, the user will be alerted that they should not drive and the message the user previously wrote will be sent to the designated contact with the user's current address (deduced from the GPS coordinates) appended to the end. This will both prevent the user from driving drunk and ensure that someone the user trusts will know of the user's location and
inebriated state and be able to come pick them up.

3 Background

3.1 Mobile Application Development

Applications developed for mobile devices are inherently different from those designed for computers. While traditional computer applications enjoy a large screen real-estate to lay out features and powerful hardware to run them, Android applications must be comparatively simpler. However, Android applications have access to a greater selection of sensors including accelerometers, GPS, and highly responsive touch-screens, providing opportunities for unique software and intuitive controls in mobile games and applications.

3.2 The Android Operating System

Android is a mobile platform designed by Google and released by the Open Handset Alliance in 2007. Built on the Linux Kernel, it was created to be a truly open operating system, and therefore the source code was released in 2008 [3]. Anyone can access and edit the source code of the operating system, allowing for adaptation to emerging technologies. Applications are equally encouraged to share in the openness of the system. They can freely access phone functionality, including GPS and text messaging, and run background processes, features less readily available on Apple's iOS system. In addition, unlike iOS applications, applications designed for Android devices do not have to go through a screening process to be made available on the Android application market. The Android Software Development Kit (SDK) allows users to employ the Java programming language to create applications to publish to the Android application market.

3.3 Previous Work

Other endeavors have been made to address the problem of drunk driving, resulting in other safe drinking applications for the Android system. Some have the user enter the number and type of drinks as they have them, and calculate their blood alcohol content based on this. Others allow the user to take a test that measures their sobriety through fine motor skills or reaction time. However, these applications require the user to remember to activate them after they have started drinking, and they merely advise the user against driving if the test is failed or the blood alcohol content is calculated to be too high [4].

Our application is unique in that the user can activate it before they begin drinking and it will automatically issue a sobriety test when the user leaves the area. Also, if the test is failed, our application is the only one of its kind that will automatically send an alert to a designated contact with the user’s location. The additional levels of automation in our application make it more likely to be used and minimize the risk of human error accentuated by alcohol. This innovative approach to a safe drinking application is a more efficient and useful model than its competing products because it requires much less participation from the user while they are actually drinking. All setup as well as the application’s activation are handled while the user is still sober.

4 Software Creation

4.1 Design

We decided to work closely with the iOS project group for the brainstorming and
design stages of our project in order to produce a cross-platform application.

Our first week was spent in the brainstorming and design process. We chose our application’s purpose, functionality, and name, and then began to design our User Interface.

4.2 Drafting the User Interface

The User Interface of an application consists of aspects with which users interact, generally through sight and touch. It acts as a middleman between the user and the application's underlying functionality and determines an application's ease of use; a badly designed, confusing User Interface can ruin even the best code.

We created a PowerPoint prototype to model our user interface, and tested it with a small test group, implementing their suggestions to make the interface more straightforward. Once users became more comfortable using the interface, we began to move on to the programming and development stage.

4.3 Implementing GPS and SMS functionality

GPS and SMS services are essential to the functionality and practical viability of the app. Both require permissions to access, and Android alerts potential users that the app uses these features of their phone before installation. SMS is accessed with an SMSManager object, which was used to write a simple SMS sending method which takes as arguments a phone number and message. This allowed for easy implementation into the more complex code of the sobriety test.

GPS functionality is accessed with a LocationManager object. When the user sets a radius and activates the app, a Proximity Alert is added to the app's global LocationManager object, with the radius and user's current location as arguments. A Broadcast Receiver was written to respond to the Proximity Alert. Thus, when a user steps out of bounds, the Broadcast Receiver is activated, ringing an alarm and vibrating the phone. If the user does not respond within one minute, or if he takes the test and fails, the SMS sending method will be invoked if the user has activated this in the User Preferences menu.

This menu also gives the option of providing the user's location, determined by passing GPS coordinates to the Geocoder class, to the contact in the text message. While the user can provide this information in his predefined text message, providing an automatically generated address facilitates the contact's task of finding the user. Therefore, the phone automates this process, and discerns his address based on his GPS coordinates. The Geocoder class is responsible for this; given the user's latitude and longitude coordinates from a Location Manager object, a Geocoder object returns his street address, which is then concatenated to the end of the text message. This functionality was tested on three different phones, running Android versions 1.5, 2.2, and 2.3.3. Although we encountered some troubles receiving correct addresses on the older phones, the address recognition worked properly with the latest edition. While it is unfortunate that users of older Android phones will therefore not be able to use this feature on their devices, users of newer Android phones can benefit fully from this feature.

4.4 Creating a Sobriety Test

An important feature in our application is the sobriety test. This is one area in which our application differs from that of the iOS project group. While they chose to use a
button press on a specific cue to test reaction time, we decided on a more complicated test. Our test, similar to common balance tests issued by policemen to suspected drunk drivers, requires the user to attempt to hold their phone in such a way that the screen is facing straight up for twenty seconds. The phone’s orientation sensors pick up how far from level the user’s phone is. How steady or shaky the user is measured by taking the standard deviation of all values read by the sensors. The user will have taken the test in advance, sober, in order to calibrate to their own standard level of balance. If the value given by the test is more than fifty percent higher than the value saved during calibration, the user is assumed to be impaired and the phone issues a text message alert to their designated contact.

4.5 Cross-Platform Complications

Several issues arose from our choice to create a cross-platform application. One issue encountered in creating the User Interface for our application arose from the issue of cross-platform consistency: should the application feature the same standard design on both Android and iOS, or should each system have its own design? We elected to use the native Android style in order to maintain consistency with the Android system and other Android applications.

Additionally, the iOS version of the application was hindered by Apple’s restrictions on iOS apps: it cannot run as a background process, and cannot send text messages automatically. These hamper the iOS version, because the application monopolizes the phone while in use and requires user input in order to text the designated contact.

4.6 Putting it All Together

Once the GPS, SMS, and Sobriety Test features were coded and running well, we implemented them into our User Interface to tie the parts together.

5 Results and Discussion

In the end, our application looked fairly different from the one created by our iOS counterparts due to different conventions of appearance between the platforms. However, we were able to keep functionality fairly consistent across the platforms. We began the testing process by testing each feature of the application separately. The sobriety test, GPS functionality, SMS functionality, and user interface were each tested independently to ensure that they could perform their tasks properly. After we were sure that the individual components worked, we began the process of putting them together. While in the Governor’s School environment, we obviously did not have access to any potential drunk drivers, which made testing the completed application more difficult. However, we did have an opportunity to test our sobriety test when representatives from the Summit High School administration came to see a demonstration of our application. We gave each of them a chance to calibrate the test and then take it again wearing glasses covered in Saran Wrap. This, in combination with them standing on one leg, left them off-balance enough to simulate intoxication. They inevitably failed the test, but when they attempted it again with both feet on the floor and still failed, we realized that the orientation sensors were more sensitive than we first thought. We made the test less sensitive after this, and it is now much more effective at measuring a person’s sobriety.
6 Figures

Figure 1: Main Menu

Figure 2: Setup Screen

Figure 3: Set Bar Size Screen

Figure 4: Set Message Screen
7 Conclusions

Our application has the potential to keep drunken people from getting behind the wheel and to make streets safer. Its simple user interface, largely automated model, and ability to run as a background process make it an improvement over existing Android safe drinking applications. Also, our decision to make the application cross platform will allow it to reach more users. Ideally, we would have had people test it in the situation for which it was designed, but we lacked access to people who were likely to get drunk. Nevertheless, we are confident that our application can work in a real world situation and stop its users from making mistakes while under the effects of alcohol, and that once released, this application will help save lives.

8 Acknowledgments

We would like to thank Christine Hung of Osmosis Learning for acting as our project advisor and helping us through the application creation process. We would also like to thank Pranav Challa, our RTA mentor, for providing helpful insights and allowing us to use his phone for testing. We would like to thank Paul Sears, Don Tobey, Stacey Grimaldi, and Ken Shulack of the Summit High School administration for helping to test our application. We would also like to thank Jean Patrick Antoine, the Governor’s School Program Coordinator, Dean Ilene Rosen, and Daniel Cobar, Head RTA. In addition, we would like to thank the Governor’s School Board of Overseers and the sponsors of the 2011 School of Engineering and Technology: Rutgers University, the Rutgers University School of Engineering, Morgan Stanley, the State of New Jersey, Lockheed Martin, Automated Control Concepts, Inc., Silver Line Building Products, Sharon Ma and Nan Yao, the Tomasetta family, and Laura Overdeck for their support of the program.

9 References

http://www.centurycouncil.org/learn-the-facts/drunk-driving-research

[2] The Federal Aviation Administration, “Alcohol and Flying, a Deadly Combination”

http://www.openhandsetalliance.com/android_overview.html

https://market.android.com/