No Text-Drive App

Tejas Bhoir, Mithulesh Kurale, Brett Lechner, and Prithvirajan Venkateswaran

{tsb88, mk1315, bgl30, pv137}@scarletmail.rutgers.edu

Advisor(s): Hubertus Franke, Hana Godrich, Peri Akiva

Goal

- ❑ To create a seamless and user friendly Android Application that helps eliminate distracted driving by disabling notifications.
- ❑ App must automatically block notifications, send auto-replies, and be able to be temporarily or indefinitely disabled.
- ❑ The app must be able to distinguish when the user is actively driving, sitting at a red light, or sitting in the car while parked.

Motivations and Objectives

- ❑ Motivations
  - Texting while driving increases the likelihood of an accident by 23 times compared to driving under normal circumstances.[2]
  - The problem has only become more frequent as the usage of social media apps increases.
  - About 50% of people between the ages of 16 to 24 have admitted to texting and driving in the past.[2]
- ❑ Objectives
  - Create an application that is easy and convenient to use for the user.
  - Noticeable reduction in the number of users who text and drive.
  - Application has a lasting value that encourages and spreads its use.

Challenges

- ❑ Battery Usage[3]
  - Figuring out the best method to approximate a vehicle's speed without consuming too much of the phone's battery.
  - Research on whether to use GPS coordinates, the accelerometer, or a combination of both for the best optimized battery usage.
  - Battery intensive apps deter usage.
- ❑ Activity Recognition API[1]
  - Getting an accurate prediction of the users activity based on device movement measurements.
  - Implementing the API within our service that enables/disables the notifications.
- ❑ UI Design
  - Creating a UI that allows the user access to everything he/she would need while not overcomplicating the menu.

Acknowledgement

We would like to thank our advisor Dr. Hubertus Franke, Dr. Hana Godrich, and Peri Akiva for their continued support throughout the project.

Methodology

Using Bluetooth

User selects vehicle from Bluetooth list → App stores the MAC address of the vehicles Bluetooth. → App starts a service to check for vehicle Bluetooth anytime the phone connects to a device. → App approximates the speed using GPS → App disables notifications when the vehicle Bluetooth and speed condition are satisfied.

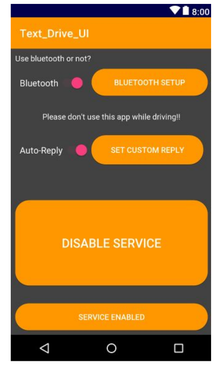
Using Activity Recognition API

App starts a service to check if user is driving using the Activity Recognition API. → App approximates the speed using GPS → App disables notifications when the user is determined to be driving and the speed condition are satisfied.

- ❑ How speed is calculated
  - The speed of the vehicle is obtained by taking the three most recent pairs of GPS coordinates and finding the average distance between them and dividing that distance by the time interval between each GPS ping. This value is then converted to km/h.

UI Design

Initial UI (debugging)



Final UI (User Friendly)

Results/Conclusions

- ❑ We were able to accomplish all of the core functions of the application such as detecting Bluetooth connections, tracking the vehicle's speed, and disabling/enabling notifications when these conditions are met/not met.
- ❑ We implemented the Activity Recognition API so that a user that does not have Bluetooth in their vehicle can still accurately enable/disable notifications while driving.

Future Work

- ❑ Parental Features
  - Allow a parent to configure the app and lock out their child from disabling the service.
- ❑ Geofencing
  - Set a geofenced location around a center point so that if a paired phone leaves this geofenced area, the parent device gets a notification.
- ❑ Improved Battery Consumption[3]
  - Use a combination of GPS and other sensors such as the accelerometer to obtain the vehicles speed to consume less battery.
- ❑ Improved Performance
  - Implement thread handling so that many of the various functions of the app work in parallel and use less of the CPU and memory.
- ❑ Improved UI
  - Create a cleaner and simpler UI to improve use and functionality.

References

[1] https://developers.google.com/location-context/activity-recognition/
[2] Richtel, Matt. "In Study, Texting Lifts Crash Risk by Large Margin." *The New York Times*, The New York Times, 27 July 2009, www.nytimes.com/2009/07/28/technology/28texting.html.
[3] https://developer.android.com/guide/topics/location/strategies.html