

Emergency Pocket Placard: Providing Emergency Response Information via Mobile Application

Chicora Chandler, Jotham Greer

{skyyetece, jotham.greer}@yahoo.com

Loretta A. Moore, Ph.D., Jacqueline M. Jackson, Ph.D., Advisors

{loretta.a.moore, jacqueline.m.jackson}@jsums.edu

Jackson State University, Jackson, MS 39217

601-979-2105

Nitin Khanna, Ph.D., Edward J. Delp, Ph.D., Advisors

{khannan, ace}@ecn.purdue.edu

Purdue University, West Lafayette, Indiana

ABSTRACT

The purpose of this project is to develop an Android application to assist first responders in effectively responding to dangerous situations such as wrecks that involve hazardous material (HazMat) carriers. The goal is to provide a user-friendly Android based application to automatically identify the threat level and provide response information in such a situation. We are utilizing the most recent Emergency Response Guidebook to assure that the application's database is reliable and up-to-date. Additionally, to make the application network-independent we are using Android's SQLite Database for data storage and we will also use MATLAB for prototyping the initial image processing system for the HazMat placards. Lastly, we will bring all of this together using the Android Software and Native Development Kits (SDK and NDK) to create the application. The project will be completed in two phases. At the completion of the first phase, we expect to have an application that is capable of receiving user input about the HazMat placards such as class number, ID number, etc., displaying images of placards matching the user's specifications, and displaying emergency response information on a selected placard. Phase two will include adding the feature to allow the user to simply take a picture of the placard and having the application automatically process the information on the sign.

Categories and Subject Descriptors

D.2.10 [Software Engineering]: Design - *Representation*

General Terms

Design

Keywords

Android, hazardous material (HazMat), placard, identification.

1. INTRODUCTION

Human injury and property damage have been caused by the hundreds of thousands of incidents involving hazardous materials carries that have occurred over the past ten years. By having an application with the ability to provide a readily available accurate representation of the response information on the spilled material, first responders at the scene will be able to assess the situation in a quicker, easier, and more efficient manner. With the use of Android software development tools, along with the use of image processing, this application will be brought to life. Current development is focusing on accessing the information by allowing the application to accept various inputs from the user, however in the future, a camera feature will be added to retrieve all needed information from a simply taken image.

2. HAZARDOUS MATERIAL PLACARDS

Hazardous material placards or HazMat placards are signs regulated by the United States Department of Transportation that are required to be used when shipping hazardous materials cargo and dangerous goods in the United States. HazMat placards are diamond-shaped and have some type of symbol at the top of the placard and a class number at the bottom. Additionally, HazMat placards sometimes have the 4-digit UN/ID number directly on them and sometimes the UN/ID number can be found on a separate panel. Currently, the United Nations are implementing a process called global harmonization in an attempt to regulate hazardous material placards around the world.



Figure 1. Examples of hazardous material placards and placement.

3. EMERGENCY RESPONSE GUIDEBOOK

The Emergency Response Guidebook (ERG) is primarily a guide to aid first responders in quickly identifying the specific or generic hazards of the material(s) involved in the incident, and protecting themselves and the general public during the initial response phase of the incident. This book categorizes thousands of hazardous materials by their class number, ID number, and their name. Additionally, all of the materials are grouped by 3-digit guide numbers which leads to the appropriate response information for those materials.

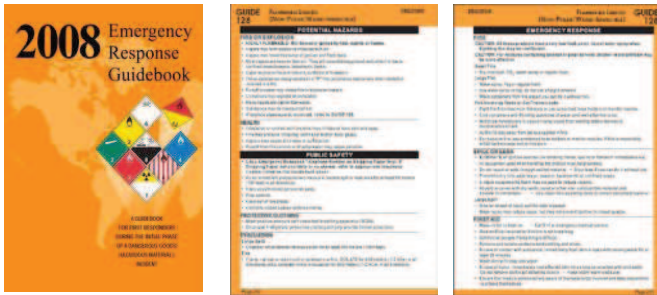


Figure 2. Cover and actual pages from of the 2008 Emergency Response Guidebook. This is the most recent version of the Emergency Response Guidebook.

4. ANDRIOD

Android is a software stack for mobile devices. It consists of its own operating system, middleware and key applications to be used with its included software development kit (SDK) and tools, along with the application processing interfaces (APIs) to begin the development of applications using the Java programming language. Some of the many features include the application framework, Dalvik virtual machine, integrated browser, optimized graphics, SQLite, media support, GSM Telephony, Bluetooth, EDGE, 3G, Wi-Fi, Camera, GPS, compass, accelerometer, and a rich development environment including a device emulator and debugging tools. A set of C/C++ libraries used by various components of the Android system are exposed to developers through the Android application framework.

Android's effective runtime has many contributing factors. A set of core libraries also provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, which has its own instance of the Dalvik virtual machine. This allows the device to run multiple virtual machines efficiently. The files are then executed in the Dalvik Executable (.dex) format to achieve minimal memory footprint. The virtual machine runs the classes that have been transformed into the .dex format and compiled by a Java language compiler. The Dalvik virtual machine depends on the Linux kernel for core functionality such as threading and low-level memory management.

Android's primary system services such as security, memory management, process management, network stack, and driver model relies on Linux version 2.6. The kernel also behaves as an abstraction level between the hardware and the remainder of the software stack.

5. SOFTWARE & NATIVE DEVELOPMENT KITS

The Android Software and Native Development Kits are both tools provided by Google to assist with application development for the Android platform. For this project, the Android SDK plug in for the Eclipse Independent Developer Environment (IDE) was used. The software development kit played an essential part in the development and design of the user interface. The native development kit is what allows embeddable components that make use of native code. The NDK also includes tools and build fields that can be used to generate native code libraries from C and C++ sources and even a way to embed these libraries into application package files. This can provide benefits to certain classes of applications, in the form of reuse of existing code and in some cases increased speed.

6. USER INTERFACE

The user interface of the application was constructed to ask the user questions about the placard in order of the relevance of the information. Initially, the application will prompt the user to enter the 4-digit UN/ID number. This is the first prompt due to the fact that all necessary response information is instantly retrievable with the UN/ID number.

In the event that the user doesn't provide a UN/ID number to the application they will then be prompted to enter the class number on the placard. After that, the user will be presented with a grid of colors and asked to select the color that corresponds to the color of the placard that they see. Once a color has been selected the application will take all of the information previously entered by the user and use it to generate a grid of placards that match that information. Finally, the application will display a dialog box with any information that is immediately prevalent over other information and then the user will see a screen with the response information separated into subsections that can be accessed by clicking the corresponding button.



Figure 3. Layout for our HazMat Identification Application.

7. SQLITE DATABASE

SQLite is a powerful and lightweight relational database engine available to all applications. It stores structured data storage in a private database. SQLite is a popular choice for the database engine in mobile devices for reasons such as: it has a small code footprint, makes efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator.

The Emergency Pocket Placard's database will contain three different tables to organize the data. The GUIDE table will be the primary table. It is the most important table because it will contain the numbered guide pages which contain all of the emergency response information. The entities that will be included in this table will be the guide number (primary key), fire or explosion, health, protective clothing, evacuation, fire, spill or leak, and first aid.

Guide

G num	fire Ex	Hlth	Saf	cloth	EV AC	ER fire	ER spl	F AD
118								
119								

Figure 4. Layout displaying the Guide Table.

The second table will be the ID table. It will be used to hold all id numbers and material names. The entities of this table will be the id number (primary key), material name, guide number, and table. Lastly, the NO_ID table will be created. This table will hold information essential to identifying placards when the id number is unknown. The class number (one portion composite primary key), color (another portion of the composite key), symbol (last portion of the composite primary key), and guide number.

The ERG will be used to populate each table. The color-bordered sections of the book will be imported into its corresponding table. The orange-bordered pages will be used to populate the GUIDE table. The yellow/blue -bordered pages will populate the ID table. The NO_ID table will not be populated by a color-bordered section of the guide, but by the information from the table of placards and initial response guide to use on-scene.

In order to import the ERG guide into the database, the ERG can either be completely manually imported, or by using text files and an .IMPORT line command. We were able to convert the entire ERG book into a text file using an online tool called the Adobe PDF Conversion by Simple Form. After the book was converted into a text file, the file had to be divided and saved into two separate files, one of the yellow-bordered pages, and the other of the orange-bordered pages. Next, the pages have to be formatted in order to be automatically imported into the tables. We were able to go through and began manually formatting the yellow-bordered pages, however we ran into difficulties when attempting to add separators. The file is much too large to manually add separators into each blank space. To address this

issue, we are exploring the use of the vim editor to search and replace special characters with separators. Once the files are correctly formatted, we will be able to easily import this information into the database. The table of placards and initial response guide to use on-scene that will be used to populate the NO_ID table will have to be analyzed and manually converted into a text file to also be automatically imported.

8. MATLAB & IMAGE PROCESSING

In the future, the Emergency Pocket Placard (EPP) will allow a user to take an image as an input, interpret the information within the sign, and return the output. However, a couple of steps must be taken. Along with capturing an image, the following steps include image/segmentation which involves of detecting a placard within an image. Second, we must be able to recognize the content of this placard by detecting its color, ID/class number, and symbol. Next, the translation of these contents will be applied. As a result, that information will be the user's output. This goal will be accomplished through a task called image processing which is done by using MATLAB.

The figures below demonstrate how image processing is done. First, the original image is converted to a gray-scale image. In other words, MATLAB stores this image an individual matrix, with each element of the matrix corresponding to one image pixel. Next, thresholding is performed on the gray-scale image which minimizes the intraclass variance of the black and white pixels, then computes a global threshold (level) that can be used to convert an intensity image to a binary image. Afterwards, canny edge detection is used to convert the binary image as its input, and returns a binary image of the same size as the original image, with 1's where the function finds edges in the binary image and 0's elsewhere.



Figure 5: Displays an original image of a placard (white border is behind the placard) which is converted to a gray-scale image, thresholded, and filtered.

9. RESULTS

The application is still in development, however, once it is completed we expect the application to be capable of receiving user input about the HazMat placards such as class number, ID number, etc., displaying images of placards matching the user's specifications, and displaying emergency response information on a selected placard.

10. ACKNOWLEDGEMENTS

Thanks to the following people for all of their contributions to this project: Dr. Mireille Boutin, Andrew Haddada, Dr. David Ebert, Marti Burns, Tim Collins, and Mike Blann and Tobias Frost of the Lafayette Fire Department.

This work is supported by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001.

11. REFERENCES

- [1] "PHMSA - Incident Statistics" PHMSA - Home. Web. 01 Aug. 2010. <<http://www.phmsa.dot.gov/hazmat/library/data-stats/incidents>>
- [2] Android Developers. Web. 01 Aug. 2010. <<http://developer.android.com/index.html>>.
- [3] "USDOT Hazardous Materials Transportation Placards" Web. 01 Aug. 2010. <<http://environmentalchemistry.com/yogi/hazmat/placards/>>
- [4] Liang, Sheng. The Java Native Interface: Programmer's Guide and Specification. Harlow: Addison-Wesley., 1999. Print.
- [5] Downey, Allen, Jeff Elkner, and Chris Meyers. "Python for Software Design: How to Think Like a Computer Scientist." Green Tea Press: Free Computer Science Books. Web. 01 Aug. 2010. <<http://www.greenteapress.com/thinkpython/>>.