SafePath

Don’t Go There!

Interactive Prototype and User Testing

*Stephen McCarthy*
Primary Android interface designer and programmer, scenario script writer, quantitative data analyst

*Andrew Lake*
Primary Android backend programmer, secondary Android interface programmer, secondary video editor, secondary writer, scenario director, primary (non-interactive) interface designer, and user interviewer.

*Andhita Primandini*
Website designer, primary video editor, photoshop guru, Zelda (scenario actress), secondary (non-interactive) interface designer, storyboard artist, and user interviewer.

*James Hayes*
Primary writer, primary copy editor, user interviewer, inserter of silliness in the middle of papers, paper prototype manipulator, storyboard artist, Raul and Ralph (scenario actor).

SafePath empowers members of the University of Maryland community to take control over their personal safety. By providing walking directions that take into account UMD safety infrastructure and past crime data, SafePath aims to prevent crime before it happens and make it easy to have peace of mind by providing easy access to campus safety services.
Overview

As part of the greater DC area, the pervasiveness of crime has long been a stigma of College Park. Although crime rates have been decreasing in recent years, many in the UMCP community still feel unsafe when walking both on and off campus, especially at night [10]. Our proposed solution is SafePath, a personal safety application for Internet-enabled mobile devices. SafePath’s primary use will be to generate a safety heat map that shows how the likelihood of a crime occurring varies with your location on and around campus, and to use this data, along with other safety information, to create a safe route that the user can follow to walk to their chosen destination. The application primarily uses the location of safety infrastructure (such as lighting, cameras, and PERT phone locations) and to a lesser extent the locations of past crimes when determining the safety of an area. SafePath will also contain contact information for existing safety programs and a panic button that alerts police if an emergency happens. SafePath expands beyond what has been done previously by combining safety information with route generation, and also by including information about safety services that are specific to the University of Maryland.
Background
Past attempts at addressing personal safety have approached the problem in three main ways: panic buttons, escort services, and information.

Panic buttons are applications or services which are used to alert trusted contacts or emergency responders quickly while an emergency is occurring. This implementation takes many forms. Some record your attack in order to assist authorities, such as M-Urgency (Fig 3.1)[8] and SOS-Link (Fig 3.2)[3]. Others set a timer that if not deactivated or extended with a pin in the given time activates the panic button, such as StaySafe (Fig 3.3)[5]. Another approach, used by the CSU San Marcos Safety App, it to simply have a button that immediately notifies authorities (Fig 3.4) [1]. Through SafePath we attempt to prevent such applications from needing to be used by providing users with the information necessary to avoid dangerous situations.
Escort services, on the other hand, attempt to deter crimes and assist while one is occurring by providing a person to travel along with the user. This is implemented both live, with escorts who walk or drive the user to their destination (UMD Escort Service, NITE Ride) and virtually, where operators stay in contact with the user through either chat or phone call and contact authorities in case of emergencies [StreetSafe, Fig 3.5 [6]]. While these are very effective because they put into practice the mantra “safety in numbers”, live escorts are typically not available for every destination and virtual escorts, like panic buttons, cannot actively prevent the crime.

Finally there are services which provide the information that is necessary for users to make informed choices concerning their path. This information is available in multiple forms such as the UMD Alerts system, which notifies subscribers of recent crimes on campus, crime and safety information from campus police, and locations of user-reported crimes throughout the city shown on a map by SpotCrime (Fig 3.6) [7][2][4]. Most of this information is presented as raw data (such as text or sets of unfiltered geotags) which may be difficult for users to understand and utilize. SafePath will integrate much of this information into a map, making the data accessible to even those unfamiliar with campus.
Prototype Description

The latest iteration of our prototype is an Android application which demonstrates most of the intended functionality of our design. The programming was done in Java using the native Android software development kit.

The program will start by displaying a map of the user’s current location, with the “INFO”, “DIRECTIONS”, and “PANIC” tabs on the sides of the left, bottom, and right sides of the display, respectively (see Fig. 4.1). The tabs can be touched or dragged to pull them out or push them back in. The user can manipulate the map by dragging and zooming in and out in the same way that they would with a typical mobile phone map application (double tap to zoom in, pinch to zoom for both zooming in and out). The map shows a crude representation of a heatmap, where relevant points have colored circles drawn around them representing the area that they affect. Well-lit areas, and PERT phones create green circles, parking lots create yellow circles, and past crime locations create red circles. While we wanted to implement this functionality fully with a colored overlay that smoothly transitioned from red to yellow to green over the entire map we were unable to do so due to the large amount of required backend programming. The locations of parking lots and PERT phones were incomplete but mostly correct, well-lit areas were estimated, and past crime locations were (mostly) generated randomly and a few were placed based on knowledge of areas with known crime issues (e.g. along Knox Rd.)
The main idea of the design was that users would be familiar with using other map-based programs, and that ours should have a similar feel, but would differentiate itself by presenting safety information prominently. One goal was to immediately immerse the user in a content-rich display by making the map the starting point for all tasks, as opposed to starting with a menu system that eventually leads to map-based user interactions. The focus is on the implementation of only a few major functions: service information (and more specifically UMD NITE Ride), showing the general safety of an area using the “Show” functionality to jump to an area for visual inspection of the heatmap, and navigation of a safety-influenced path. Each of these is accessible very quickly from the starting screen by pulling out the appropriate custom-tailored panel (and in one case, “show”, by additionally clicking on the right tab within the bottom panel.) (See Fig. 4.2, 4.3, and 4.4.)
Fig 4.2: “INFO” interface
(Note: Due to a technical glitch the “INFO” handle is not displayed but should be; interaction with the handle is still possible, however.)

Fig. 4.3: “SHOW” interface

Fig. 4.4: “NAVIGATE” interface

In order to complete the first task, calling NITE ride, the user can touch or drag out the “INFO” tab to show a drop down menu with a list of safety services (Fig. 4.5). A service can be touched to provide information about it, and a call button is displayed at the bottom to contact that service. In the case of this task, the user would touch the drop down menu, touch “NITE Ride” and then touch the call button. For our prototype not all of the services have had their information tabs filled out.
Our second task is to find safety information about a given area. The user would begin this task by touching or dragging the directions panel handle to bring it up. In this case the "SHOW" tab is already displayed (Fig. 4.7) Touching the input box will bring up the built-in Android soft keyboard (Fig. 4.8), which the user can use to type in a building name, with assistance from autocomplete (Fig. 4.9.) Once the user hits “Go” or “Done” (Fig. 4.10) on the soft keyboard the map will jump to the entered location is centered and highlight it with a blue flag (Fig. 4.11).
The third task is to get walking directions from the user’s current location to another one. In order to do this the “DIRECTIONS” panel would be brought up again, and the use would tap the right tab.
labeled “NAVIGATE”. The user can touch and then type in locations for each box (Fig. 4.13 - Fig. 4.15.) When the user hits “Go,” walking directions will show up visually indicated by a blue line on the map to guide them to their destination (Fig. 4.16 - Fig. 4.17). In our final implementation we were unable to implement turn-by-turn navigation due to our lack of familiarity with Android listeners as they apply to location services. We planned on providing turn-by-turn directions and placing the current direction in a top panel that would appear only during navigation. The user would be able to drag down this panel to get a list of all the future turns coming up as well.

Fig. 4.13: After entering source

Fig. 4.14: Autocomplete of destination

Fig. 4.15: Post-tap of autocompletion

Fig. 4.16: After tapping “Done” or “Go”  
Fig. 4.17: Pinch-to-zoom to inspect path
Owing to time constraints, there are some limitations to our prototype. In addition to what is mentioned above, the panic button is not functional, and some buildings will not show a location when keyed in. There are also small visual errors with the “INFO” panel’s handle and the soft keyboard obscuring the view in some cases (in these circumstances the user needs to press the “Back” button to restore the view correctly). These do not detract majorly from completing the tasks, but are things that would need to be completed in future iterations of the project.

Testing Method

Our participants were MN, an elementary education major, and NJ, an accounting and finance major. We found them by approaching people at the Stamp Student Union and asking if they wished to participate in a usability study for a mobile phone application. After they agreed, we gave them a consent form to sign, and began the test.

The tests took place on tables in the food court of the union. There were few people around, and the noise level was low. Conditions were essentially the same as they were for our low fidelity prototype testing. James explained the testing procedure to participants and gave them tasks, while Andhita filmed and Stephen took notes.

At the beginning of the test, we read to participants a short speech explaining how the testing would proceed. We then told them to take a few minutes to familiarize themselves with the application, and they used this time to try out the different tabs and move the map around. Once they were ready, we gave them the three tasks in a random order. After the tasks were finished, we asked the users for any comments that they had and thanked them for participating.

The tasks were the same as the ones used in previous user testing except that the locations were changed. The first task was to call NITE-ride, which involved using the info tab and its drop down menu to retrieve NITE-ride’s information page and then hitting the call button. We considered this the easy task. The second task was to get safety information about an area, which is done by bringing up the “show” tab, inputting the location, and then examining the map. This was our moderately hard task. Our third task, which we considered our hard task, was to get directions from one point to another. This is accomplished by bringing up the navigation tab, filling in the start and destination boxes, and then hitting “go.”

During the experiment we were looking to see what errors were made by the participants, and try to discern why they made them. We were specifically looking at whether they found the correct function on their first try, where they went if they did not go to the correct place, and how long it took them to figure out how to complete the task. We were looking at how quickly and confidently they navigated through the tasks, and asked questions afterward, to see if they found task completion easy or were muddling through but made correct choices. Overall, we were mainly relying on qualitative judgments to assess how well the interface worked, due to the low number of participants.

Testing Results

The participants’ reactions and the results of the testing were both mostly favorable towards our prototype. Both participants were able to complete tasks 1 and 3 with little or no difficulty, and all tasks were either completed or failed quickly. In addition, the participants said that they liked the interface and found it easy to use.
The major problem came with completing task 2, and that was because they did not understand that the colors on the map represented the level of safety. We were already aware of this problem, but had not had time to implement a starting dialog box which explained the meaning of the map coloration. Since they were seeking information, they pulled up the info tab first for task 2, but were then unsure of how to proceed. After this, participants correctly pulled up the “show” tab and reached the point where the location was visible on the map, but then were unsure of what to do, ultimately deciding that they should open up the info tab and call the police to find out about an area. There was clearly some confusion about the usage of the info tab, as also during the trial period the participants would sometimes hit the info tab, and then they appeared to be hesitant, as though they did not see what they were expecting. On the other hand, they did go to info and immediately find NITE-Ride when their task was to do so.

The testing was successful at exposing areas that could be improved. NJ suggested that we should implement some sort of method for touching a point on the map to bring up a context menu for things that you could do there, such as navigating or centering on that location. While a context menu may not be necessary, some sort of function should be activated when a point on the map is touched. Also, when he was attempting to input “A.V. Williams” the autocomplete did not recognize “av” and he had to correct it to “a.v.” in order to make it work. This suggests that improvements could be made to the autocomplete. Another issue was that MN pulled out the panic tab at one point and had trouble getting rid of it, presumably because the touchable area was too small. She also had trouble with pinch to zoom because the screen had become crowded and she was trying to avoid hitting the open tabs.

Further Interface Revisions

There are a number of changes that we would make to the application in the next iteration. At the suggestion of user NJ, we would implement a context menu that appears when the user touches and holds a point, as he felt that this was more natural for navigating, and we also believe that adding this would be helpful to users. The names of the tabs need to be changed, as the names of the “directions” and “info” tabs caused problems for users. One possibility is changing “info” to “services” and “directions” to “locations,” but it is difficult to eliminate ambiguity here and additional testing would need to be done to determine the best names. We would also make the tabs a bit thicker in order to make them easier to hit. Lastly, it would be great if our current location indicator, a red turtle shell, sprouted legs and a head and moved across the map in tandem with the user walking along their route when a path was being followed.

In addition to the design changes, there are a number of technical issues that hindered the effectiveness of SafePath. We were not able to implement a high fidelity heatmap given the timespan of the project, and the lower fidelity one proved confusing for users. The tutorial on the meaning of the heatmap was not considered very important during implementation, but we now realize that it would be crucial to put it in any further project iterations. There were also problems with the keyboard appearing at incorrect times, and the info tab disappearing when pulled. In addition, the autocomplete was usually effective, but had room for improvement, and the same could be said of the pathfinding algorithm. All of these issues are fixable, and simply require additional time to work through.
**Discussion and Lessons Learned**

The process of creating and testing the interactive prototype has been the most difficult of the project so far. Implementing it was much more difficult than expected, but important things were learned. First, changing from paper to digital created implementation problems which meant that the design could not be fully realized, at least initially. Also, we believe that the testing results were heavily influenced by the fact that the prototype is now on an actual phone. With the paper prototype, our users were very slow and deliberate, carefully thinking about and making each selection. With our new prototype, the users were much more natural with it, quickly picking it up and hitting buttons so quickly that it was at times hard to follow. Thus, while the paper prototype was good for getting some ideas, there are aspects of the design that cannot be tested this way, as the decision making process is much quicker and less deliberate.

One improvement to future usability testing would be to do testing of certain design aspects in isolation. The importance of the heatmap tutorial and the naming of the tabs was not fully appreciated until this prototype testing had been done. It is more difficult to assess the usability of the interface as a whole because it was so heavily influenced by these factors, which are only a part of the project. It may have been helpful, and would be helpful if we were to continue, to conduct smaller user tests that focused only on what tab names were appropriate, or which tab users would pick if asked to perform a certain action.

Overall, we believe that we have had many successes at developing our application. We spent a lot of time interviewing users and experts about how best to implement a personal safety application. The final design is as simple as it can be, with only 3 tabs, each leading to important functionality in only a few touches. While some problems were encountered when translating it into a functioning prototype, these could be fixed, and we believe that the general design is good. Its implementation was technically challenging, and we have a working, if somewhat buggy, final product.

There were things in our design process that could have been improved upon. We waited for too long to begin learning about developing for Android, and were forced to rush development in order to finish on time. It may have also been beneficial to try out more radically different designs at the beginning in terms of both visualizations and functionality.
Website Report

The website that we made is named after our product, which is safepath.us. The website was built by using a standard wordpress theme to be a static website with a little bit of modification on the header. We use GoDaddy as our web hosting client, and built the web page in the wordpress environment. Also, we used FileZilla as our FTP Client when we set up wordpress on the computer. The web banner is made by using a combination of adobe photoshop and illustrator. We used html5 and css3 when we modified the theme that is provided by wordpress.

For the contents of the website, we put all the deliverables and the videos that we have on our course webpage into our website along with the contact info for each team member. Besides that, it also has a paragraph about our motivation for building the SafePath Android Mobile App.

We designed the website to be very simplistic and mainly focus on the content of the project. We want for the user to be able to distinguish each part of the content easily so we made different tabs for each part.
References

February 17, 2012 from: http://www.umpd.umd.edu/RECORDS/UniformCrimeReports.cfm
alert.umd.edu/faq.php
Information. Retrieved February 17, 2012 from: newsdesk.umd.edu/universitynews/
release.cfm?ArticleID=2492
Personalized Awareness and Safety with Mobile Phones as Sources and Sinks. Yale University,
yu_urbansense08.pdf
[10] Student Reviews, 2012. Students Review :: The University of Maryland College
uid=639&f=Safety&d_short=UMC&d_school=The+University+of+Maryland+College+Park
## Appendix A: Video Consent Form

<table>
<thead>
<tr>
<th>Project Title</th>
<th>SafePath</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose of the Study</strong></td>
<td>This research is being conducted by James Hayes, Andrew Lake, Stephen McCarthy, and Andhita Primandini at the University of Maryland, College Park. We are inviting you to participate in this research project because you are a student at the University of Maryland. The purpose of this research project is test the usability of a personal safety application for mobile phones.</td>
</tr>
<tr>
<td><strong>Procedures</strong></td>
<td>You will be given an Android phone to use for the experiment. You will be asked to familiarize yourself with a mobile phone application for 5 minutes. You will then be asked to perform 3 tasks that can be accomplished with this application. The investigators will observe you doing this in order to assess the usability of the application and to find out if any errors occur in the application.</td>
</tr>
<tr>
<td><strong>Potential Risks and Discomforts</strong></td>
<td>There are no known risks associated with participating in this research project.</td>
</tr>
<tr>
<td><strong>Potential Benefits</strong></td>
<td>This research is not designed to help you personally, but the results may improve the design of a safety application and help the investigators learn more about usability.</td>
</tr>
<tr>
<td><strong>Confidentiality</strong></td>
<td>Any potential loss of confidentiality will be minimized by recording your name only on this form. Your name will not be included in the collected data. We will attempt to avoid recording your face in the video recordings, and will not present any recordings in which your face is present.</td>
</tr>
<tr>
<td></td>
<td>If we write a report or article about this research project, your identity will be protected to the maximum extent possible. Your information may be shared with</td>
</tr>
</tbody>
</table>
| Right to Withdraw and Questions | Your participation in this research is completely voluntary. You may choose not to take part at all. If you decide to participate in this research, you may stop participating at any time. If you decide not to participate in this study or if you stop participating at any time, you will not be penalized or lose any benefits to which you otherwise qualify.

If you decide to stop taking part in the study, if you have questions, concerns, or complaints, or if you need to report an injury related to the research, please contact the investigator: James Hayes jameslhayes@gmail.com 240-515-1128 |
|Participant Rights| If you have questions about your rights as a research participant or wish to report a research-related injury, please contact: University of Maryland College Park Institutional Review Board Office 1204 Marie Mount Hall College Park, Maryland, 20742 E-mail: irb@umd.edu Telephone: 301-405-0678

This research has been reviewed according to the University of Maryland, College Park IRB procedures for research involving human subjects. |
|Statement of Consent| Your signature indicates that you are at least 18 years of age; you have read this consent form or have had it read to you; your questions have been answered to your satisfaction and you voluntarily agree to participate in this research study. You will receive a copy of this signed consent form.

If you agree to participate, please sign your name below. |
<table>
<thead>
<tr>
<th>Signature and Date</th>
<th>NAME OF SUBJECT [Please Print]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SIGNATURE OF SUBJECT</td>
</tr>
<tr>
<td></td>
<td>DATE</td>
</tr>
</tbody>
</table>
Appendix B: Testing Script

Intro:

Thank you for participating in our study. We’re in the early phases of designing a mobile phone application to be used by the students, staff, and faculty of UMD. The application is designed to provide safety information about campus and the surrounding area. As part of the design process we’re going to have you take a look at it and perform some tasks with the app and give us your feedback. The purpose of doing this is to get your input on how to improve the usability of our mobile app. You’re going to be working with a mobile app that happens to be on paper. This is James he’s going to helping us today, he’s going to playing the mobile phone, and handing you the screens as you work on your task. How this works is this is your mobile phone area. You’ll be using your finger to interact with the simulated touchscreen as you would a normal phone touchscreen. If there’s an area that you would normally type in with your keyboard you’ll use this pen to write in what you would type at the keyboard. We will be giving you some time to get acquainted with the app, and then we will give you 3 tasks to perform. When you think you are done with a task, say “I’m done.”

Task Descriptions:

Easy task:
It is late at night and you are trying to get home. You have heard of a service called NITE-ride, which will pick you up and take you there. Use the application to call NITE-Ride.

Moderate task:
You intend to travel to an area, and wish to know if the area is safe. Use the application to find safety information about the agriculture shed.

Hard task:
It is late at night and you need to walk across campus. Use the application to find a route to get to A.V. Williams from your current location at the Stamp Student Union.
### Appendix C: Critical Events Log

<table>
<thead>
<tr>
<th>Participant</th>
<th>Task</th>
<th>Rating</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td>1</td>
<td>0</td>
<td>went to panic button first thing???</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>could not get rid of panic button.</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>went to info, finished, said NITE-Ride task was easy</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>went first for the info tab</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>asked if she should call police to ask them</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>went to directions tab, input into show</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>did not know what to do after the location was on screen</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-</td>
<td>went to directions</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>typed location into show first, then went back and put it into navigate</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>found route and finished task</td>
</tr>
<tr>
<td>NJ</td>
<td>2</td>
<td>4</td>
<td>went to info tab first</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td>decided he should call UMD police to find out about an area</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-</td>
<td>went straight to directions tab</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>typed in “stamp” then finished with autocomplete</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>filled in “av_williams,” but when it didn’t autocomplete, changed to “a.” and then autocompleted</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>finished task</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>-</td>
<td>went to info tab immediately</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>found NITE-ride, but keyboard was in the way of the call button</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
<td>hit back to remove keyboard and finished task</td>
</tr>
</tbody>
</table>