Sean Takafuji
Jason Len
ICS 414
Team Buffalo
Assignment: Team Project #4 Finale
https://github.com/ics414-buffalo/aloha-alert

The Team 4 Finale assignment deliverable is a single pdf document containing:

- A brief description of the implementation of at least two more warning methods/modes.
- Final total LOC count for the system.
- LOC **changed** from Team Assignment 3, as well as the LOC **added** since Team Assignment 3.
- A narrated video demonstration (**no longer than five minutes**) showing the user interface and functionality. Submit a URL that we can watch, i.e., not behind a password, in mp4 format. If you have a Mac, you can do a screen recording using Quicktime, or for Windows and OS X you can use the free https://screencast-o-matic.com/
- V&V results, including: results of usability testing of the "full up" system with at least one test user, and results of the functional testing that you described in Team Assignment. You will need to convince us that your system is correct. Hint: 10 test cases won't be enough :)
- A link to your git repository so we can check the log to see that all team members participated.

After this assignment is due you will also do team peer evaluations.

Our Warning Methods:

1. **E-mail** - Currently, we have an email system in place that will be used to send emails to all of the local (and/or national) news, television, and radio stations with an alert of the various threats and warnings This warning method is fully functional, but currently we are only sending out and email to a single test account. (This working method was shown to Vincent as a part of the Team 3 deliverable meeting.)
2. **SMS/Text** - This warning method is intended to send out a SMS text message to the general public through the emergency SMS texting system. This warning method method is fully functional, but since we do not have currently have access to this sub-system, we are using a single phone number as a test account. (This working method was shown to Vincent as part of the Team 3 deliverable meeting.)
3. **Alert Sirens (sound)** - This warning method is intended to set off an audible alarm through the Statewide Emergency Warning Sirens and the local computer system. While we currently do not have access to the Statewide We included it local computer to also ring to ensure that the person who executed the alarm also gets a audible feedback notification.
4. **Console Out** - This warning method will produce a warning method that will output to the console window as a model of a notification that is sent out to the internal (government) agencies that are involved within this system. (Governor and Mayor, Hi-EMA, Military, …) The console out model was chosen as it would allow the receiving

system to more easily accept the signal and to be able to smoothly incorporate it into their own system.

Lines of Code

Final Lines of Code:

- Our Own Personal Line of Codes:1049 lines
- Total Lines of Code for Our Entire Program (including all modules): 486,564 lines

Lines of Code at Team 3 Deliverable:

- Our Own Personal Lines of Code: 751 lines
- Total Lines of Code for Our Entire Program (Including all modules): 597669 lines

Video Link:

Youtube video link: https://youtu.be/kxvil_2FdUo
Link also updated in the README.txt file on GitHub.

Verification and Validation (V&V):

➢ "Validation. The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with *verification*."
  - ○ Accuracy and Precision
  - ○ Repeatability
  - ○ Reproducibility
  - ○ Limit of detection
  - ○ Limit of quantification
  - ○ Curve Fitting
  - ○ System Suitability

➢ "Verification. The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with *validation*."

- **Brute Force Testing:** Testing of all clickable links and testing the entries of the following to text boxes.

  We are working on using the "**brute force**" method of testing for the majority of the site. We choose the brute force method due to the fact that the majority of our system is a point and click system of buttons, there is very little other user input. Because most of the input was button related, we felt that it would be possible to thoroughly exhaust all possible test cases for them. From there, we only a couple of input text boxes, which needed testing, to do this we ran through a series of test cases. Here is a listing of the different user actions that were tested.

  ✓ Clicking on "whitespace" area that are not clickable
  ✓ Main Page

  - ✓ Amber Alert
  - ✓ Natural Disaster
  - ✓ Foreign Threats
  - ✓ Cancel Threats
  - ✓ Multiple Different Whitespace Clicks (10+)
  - ✓ *Main Page” Link
  - ✓ “Alerts” Link
  - ✓ “How to Use” Link
- ✓ Foreign Threats Page
  - ✓ Missile Attack REAL
  - ✓ Missile Attack TEST
  - ✓ Cancel Button
  - ✓ Multiple Different Whitespace Clicks (10+)
  - ✓ *Main Page” Link
  - ✓ “Alerts” Link
  - ✓ “How to Use” Link
- ✓ Real Missile Attack Page
  - ✓ <CR> (we just hit enter)
  - ✓ “CONFIRM” correct response
  - ✓ “confirm” Slightly wrong answer
  - ✓ “onfirm” Slightly wrong answer
  - ✓ “confir ” Slightly wrong answer
  - ✓ “verify” Random word
  - ✓ “?” Hacking the system
  - ✓ “#” Hacking the system
  - ✓ “1=-1” Hacking the system
  - ✓ “1=1” Hacking the system
  - ✓ “asldfkj” Random characters
  - ✓ Missile Attack TEST
  - ✓ Go Back Button
  - ✓ Multiple Different Whitespace Clicks (10+)
  - ✓ *Main Page” Link
  - ✓ “Alerts” Link
  - ✓ “How to Use” Link
- ✓ Test Missile Attack Page
  - ✓ Textbox Entries
    - ✓ <CR> (we just hit enter)
    - ✓ “CONFIRM” correct response
    - ✓ “confirm” Slightly wrong answer
    - ✓ “Confirm” Slightly wrong answer
    - ✓ “onfirm” Slightly wrong answer
    - ✓ “confir   ” Slightly wrong answer
    - ✓ “verify” Random word

        - ✓ “?” Hacking the system
        - ✓ “#” Hacking the system
        - ✓ “1=-1” Hacking the system
        - ✓ “1=1” Hacking the system
        - ✓ “asldfkj” Random characters
    - ✓ Missile Attack TEST
    - ✓ Go Back Button
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Missile Attack Emergency Message Sent
    - ✓ Main Menu Button
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Hawaii EMS (from Cancel before sending) Page
    - ✓ Return to Messages
    - ✓ Alert from List
    - ✓ Main Menu Button
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Missile Attack Warning Cancelled
    - ✓ Main Menu Button
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Amber Alert Page
    - ✓ Cancel Threats
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Natural Disasters Page
    - ✓ Cancel Threats
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link

- ✓ Tsunami Page
    - ✓ Cancel Threats
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Hurricane Page
    - ✓ Cancel Threats
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Test Missile Attack Page
    - ✓ Cancel Threats
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link
- ✓ Warning Successfully Sent Page
    - ✓ Multiple Different Whitespace Clicks (10+)
    - ✓ *Main Page” Link
    - ✓ “Alerts” Link
    - ✓ “How to Use” Link

- **Testing Scenarios:** Before users was asked to run through the script, the user was given time to explore the system until they felt comfortable with the system. (No time limit.) From there, this following script was read to them in order by a moderator. The next instruction was only read after the user stated that they had completed each set and was ready to move on. No questions were to be answered by the moderator, but the moderator could be asked to repeat the instructions as many times as needed. Note: Unfortunately, time was not recorded for the completion of each instruction or for the overall performance, as speed was not considered an area of concern at the time. Looking back, this data should have been taken anyways. A listing of the follow up questions used to spur thoughts and discussions is listed below in Appendix A. Here is the list of operations in the testing scenario:
    - ○ This is a **test case.** We want to test the system with an inbound missile test.
    - ○ This is **REAL.** Send out an amber alert for Casey Brown, 12 yo, no picture, with a message of “ gone missing at Ala Moana Shopping Center food court at 1:25pm”.
    - ○ This is **REAL.** Send out an amber alert for Jamie Williamson, 5 yo.
    - ○ This is **REAL.** Send out a Tsunami warning.
    - ○ Display a listing of current warnings
    - ○ This is **REAL.** Cancel Tsunami warning.

- This is **REAL.** Cancel amber alert for Jamie Williamson
- This is **REAL.** Send out a inbound missile warning.
- Display a listing of current warnings.
- Cancel missile warning.
- Display a listing of current warnings.
- End of Test Case.

Test Users Results

User 1: All tasks correctly completed. In using the system, the user said that they found the system easy to use, but a bit “childish” or “old fashion” due to the simplicity of the pictures. As follow up to this comment, the user was asked if this factor negatively impacted the use of the system (i.e. made it harder to understand or was distracting.) The user replied that it did not, but that instead it made the user feel that the system was older system and not recently created or updated. When asked for other feedback, this user had none. It was then explained that the reason for such simple pictures was that we wanted to focus on function and did not want to have nice complex pictures that might be misinterpreted. We therefore choose to go with icon based instead of pictures that had too much within the picture (i.e. color, design, …). We wanted simple.

User 2: User 2 encountered a small problem when attempting to send out both of the Amber alerts. The typing of the text inputs were a bit difficult for the user, they kept making simple typing errors and needed the directions to be repeated (and spelt out) quite a few times. But in the end, they got all steps done correctly. (It was for this user that we would like to have gone back and have timed this test case…) When finished, this user talked about the difficulty in taking the directions about the Amber alert verbally and wished it was given in written form. The user was informed that in reality, we did not actually know how the information is presented to the “real” users, but would believe that it most likely come through written form over some type of electronic means. We apologized for the confusion for the test scenario. Besides the frustration from needing to ask for the Amber Alert scenario to be re-read multiple times, this user felt the rest of the system was pretty easy to use. Because there was time, we actually changed the email to this users email, so they were impressed when they actually received the email own their own device and saw the text message on our phones. They did not realize that these warning methods were actually fully implemented. On a closing note, this user really appreciated that there was a consistent structure of form throughout the entire project. After running through their explorations, they felt that they know where everything was, for example, “the real threat was always to the left and the tests warnings are always to the right.” They said it made them more comfortable knowing that in a real threat that they would know where to go and feel confident while doing the required tasks.

User 3: This test user was a little pressed for time and did not elect to spend much time in practice with the system before starting. When using the system, this user did not appear to have any difficulties with most of the procedure, but was not easily able to find the listing of the messages sent. This led to them taking an extraordinarily long time in trying to cancel the Tsunami Warning. When canceling the second warning, things went a lot smoother. (After this test case, we also added some more clear directions about cancelling alerts.) This user also commented about the use of icons based images being questionable and though that the icons should have been real pictures. But when asked for further explanation on impactfulness the pictures, their only concern was that they didn't "look nice", they had no functional issues (such as the pictures being confusing or distracting).

Appendix A – Follow-Up Questions

- How easy or difficult was it to determine how to proceed in each scenario? Why?
- Which was easier to understand, the pictures or the text? Or were both necessary?
- Was it easy to understand what the pictures where trying to convey? Would you have been able to figure that out without the text? Which pictures do you think need improvement, and how?
- Was it easy to understand the text? Was there too much or too little text for each situation? Which texts need improvement and how?
- Did the colored backgrounds to each button help or hinder the process?
- Do you believe that after proper training user would be able to make a mistake on this system due to some type of confusion? (i.e. such as mistakenly thinking one button was another?)
- How likely do you believe it is that an error would happen due to an accidental slip such as a user unintentionally clicking or typing the wrong thing. NOT due to the lack of understanding? (i.e. Wanting to click on one thing but accidentally clicking the wrong item?)
- How easy was it to type in the correct answer to send the warning message?
- Why do you believe that the program would require the user to type in "CONFIRM" instead of just clicking a button labeled send?
- How likely would it be for a person to accidentally send a real message in a practice situation?
- How do you feel after looking over this program simulation?
- What went well?
- Was the system easy to use?
- Did you like it?
- What could be improved upon?

Our Git Repository Link: https://github.com/ics414-buffalo/aloha-alert

Please read the README.txt file for instructions on downloading and running the necessary files to access the system.