User Testing Report

NG911

Team Members:

Rabaa Abdelrahman,

Li Pan,

Renz Rivero

April 9, 2022

Why test with users

Any system should undergo many types of tests to prove itself valid and ready to be used. One type of testing is user testing. Most often we “software engineers” will not fully grasp the idea of putting ourselves in the user’s shoes, as we are not really users.

Scenarios

User testing in our project will depend on two different scenarios. One scenario is the 911 dispatcher getting an actual call, the other is the user calling the emergency. The user testing was partially done during the planning phase, when we met an RCMP officer each week to make sure our design makes sense, as well as other people in the emergency area.

First test - sos app

The real user testing was done when the first MVP got into place. The first MVP was our application with a simple connection with the dispatcher’s dashboard. Video/audio calls were working but not reliable. We set a meeting with the officer “John Leitch” to lend Rabaa’s phone for 3 or 4 days to test the features as a ‘user in emergency’. We agreed with the officer to call us before doing a call so that we can have the dashboard ready to answer him. When Mr. Leitch was ready, he notified us that he would make an emergency call so that we could open our dashboard.

First test - sos app - results

We didn’t get any call on our dashboard. We made sure that the version he had on Rabaa’s phone was the latest stable version with MVP 1 loaded onto it, the only logical deduction was that he forgot to connect to the internet. When Rabaa got the phone back, he made sure that this was the only problem.

Mitigation

For real situations, the users in emergency would not be calling the technical support to explain to them that they need an internet connection to do the call, so we didn’t mention it Mr. Leitch, instead we got the phone back and arranged another meeting with him with a “modified” version.

We added a wifi-connectivity signifier on the homepage of our application to signify the connectivity state of the user's device, and we sent it back the Mr. Leitch for another test.

Second test - sos app

Mr. Leitch this time made sure to connect to wifi. The emergency call was perfect, and Mr. Leitch even gave us more comments on what could be enhanced this time. His advice was that everything is perfect and made sense, except the order of the call page.

Second test - sos app - results

The suggestion was to start the call with the audio enabled as a default of the chat, and the user can switch between the audio, video, and chat as convenient. Dr. Morgan later added that a user under threat may not want to have a voice call which may grab unwanted attention to him/her, so we can add another functionality for the user to mute and unmute the dispatcher when needed.

Mitigation

The package we used "web-rtc" had its own operating logic, we wanted to restructure our design to make it possible. Before rewriting the code, we had to fix the code written to make sure we fully understood the logic of the video calls.

Dispatcher test - inexperienced users

The third test was done on the dispatcher. Unlike the previous user test, the dispatcher dashboard was tested by random students at the University of Regina. The students had no training and no qualifications but their connections with the developers in the project. 2 students were picked for the dashboard, and left alone as we took our mobiles and moved away. They understood what the red line "new locations" is for and that the black line is for the previous locations. They responded to the chat once we sent a text and everything seemed to work perfectly. The students were asked to end the call, and they didn't find any difficulties doing it.

Dispatcher test - Instructor and Officer

Multiple calls were tested one after the other on the same dashboard, the second test was multiple calls with multiple dashboards. These tests were done in front of officer Meitch and Dr. Morgan. In the first test we found out that the locations sometimes did not change after the call ended. The second test sometimes worked, and sometimes showed red routes showing up and disappearing randomly.

For the first test we made sure the lifetimes of our variables function as intended, and the map widgets are refreshing each 5 seconds.

Future tests

As this report is being written, more tests are going to follow to assure stability of the system.