Introduction (Jack)

Hello Tim and Candace, thank you guys so much for joining us for our presentation. My name is Jack, with me today are John and Zain and we are formally known as the FireForce.

Problem, requirements, & framing (Jack)

Problem definition:

Candace, when you came on the first day, passionately expressing your experience, your motivation, and your goal, we knew we had to create something that had to match that energy. To do that, we first had to identify the problem we are trying to solve. After creating an affinity diagram and an empathy map of you and what we defined as our north star customer, the children, we came up with two concrete problems.

The first one is that we need to encourage the children to participate in fire safety and retain that knowledge for real-life scenarios. Secondly, whatever we design needs to be a modern approach in terms of technology and communication methods. Our rationale is like what you said, most fires that happen today are preventable. We need to create something that will allow the RFPS to save more lives by providing you with a modern fire safety application. These things quickly became our vision and the very foundation for this year's project.

From lo-fidelity to hi-fidelity prototypes (Jon)

Lo-fidelity

These are some of the sketches we drew for our envisioned app. We took inspirations from the lock screen and dial screens of iPhones to simulate the process of unlocking the phone and dialling 9-1-1.

When proposing our ideas Candace told us that our priorities should be the call simulation and teaching the parts of an address.

Some insights we were given is to use graphics to point out important parts whenever we can. This is later implemented in our house address screen as we point out where the number of the house is.

For an app designed for children, feedback is very important. We were suggested to use audios and visuals for positive reinforcement. We took this another step further into our design. We only allow the user to proceed when the correct button has been pushed. We use different feedback mediums to notify the user about the state the app is in. Then play cheering sounds and confetti to celebrate their success.

Hi-fidelity

As we were coding we realized that there will be some cultural constraints in the app. Since our app will be mainly used by children in Regina we would restrict the app to only dial 9-1-1 the North American emergency number.

We were told that children live in a make-believe world. Providing hints from a cartoonish dog and displaying the process like a game would greatly increase the interactivity of our user. To implement this we would take the concept of Clippy from Microsoft Word and have our dog character walk through the steps to dial 9-1-1.

A critique that Candace has given us is to use symbols to metaphorically visualize each function in our main screen. We thought this would be great to communicate using conceptual models. Not only will the main menu look more appealing, but it will help strengthen our signifiers.

Lastly, we proposed to display information of the 9-1-1 call on a PowerPoint presentation on an external device. As we thought that the children will have the phone screen pressed on their faces. This way the children will look like they are in a call while still being coached on what to expect in a 9-1-1 call. Candace suggested that there should be a way to access the script within the app so we decided to implement it on the call screen.

Solution Demo (Jack)

When launching the app we are first greeted by a loading screen accompanied by a cute dog and a little information blurb about the app.

When the loading is finished, we come to the home screen of the application. As you can see, we highlight areas of the screen and tell the user what each section means. First, we have Safety Tip of the Day where it displays a little info blurb with very useful tips when it comes to fire safety. Then we got 4 clickable tiles that will lead the user to a different section of the app depending on what they click.

Clicking on the house tile, it'll lead us to a brand new page where it tells the kids how to find their house number. Then, clicking on the arrow at the bottom, it'll lead us to the next page where it explains how to find your street name. Combine the house number and the street name and you got an address!

So, going back to the main page, we'll go ahead and click on the 911 phone tile.

Once again, it highlights an area of the screen with text to explain what to do. In this case, we want the kids to tap the emergency button. Now, in case the kid tapped too quickly and skipped that hint or didn't pay much attention to it, we added a little dog at the bottom of the screen that tells the kid what to do. We added signifiers like the moving finger to also signify the user on what their next objective should be.

Once we tap the emergency button, we come to the calling screen with the keypad highlighted. Now the kid should know that the keypad is clickable and they should dial a number to call someone. Another feature we added is the ability to hide the hint dog as we might want the children to attempt without any guidance.

Next, we'll try to dial the number 9 - 1 - 1, let's say I somehow forgot what number 911 starts with and I click a wrong number… we have some signifiers displayed to the children that they are indeed wrong. Thus adding the constraint that the children must input the correct number in the correct sequence to proceed to the next stage.

I'll now dial the number 9, and as you can see, it shows what the current phone number is along with cheers and confetti all over the screen to signify that this was the correct number.

I'll go ahead and tap two 1s to complete the 911 dial.

Now, the kid is prompted to tap on the phone, an affordance for calling someone, and see if their call goes through. We added the constraint that every other button is disabled except the phone button, therefore the kid can tap all they like but they have to tap on the phone to proceed.

Voila, we are now talking to the 911 operator. You'll also notice this button on the bottom right corner named script. As the name suggests, when you click on it, it'll lead you to a new page where it displays the script for both the firefighter and the child.

This concludes the demo for our app's first release. There are many more features we would love to add but due to the time constraint and the world in chaos, we decided to cut it right here. As a wise man once said on zoom, don't let perfect the be enemy of good, and we definitely don't want to bite off more than we can chew.

Technical Information

As you guys can see from the demo, we took a lot of inspiration from our Lo-Fi and Hi-Fi design as well as trying to follow our user story map with regards to the activities being done. From finding out your address, all the way to dialling and calling the emergency responders.

Now, Candace, feel free to take a quick break or grab a beverage as I quickly go over some technical details.

Our application is developed entirely through Android Studio using the Flutter SDK as well as the Java SDK for APK compilation.

Flutter is Google's UI toolkit that builds applications for mobile, web and desktop from a single codebase. One of the best features is that it has this function called hot reload. This function saved us a lot of time and what it does is that it allows the developer to make changes in the code, and when he saves the code in the compiler, it'll automatically be injected into the emulator. Meaning when you change something, you can see the change you made right away without having to exit the app and reload it again.

Developing a flutter app is quite new to us and we did come across some issues as we were developing the app. Mostly concerning the formatting. When we first developed the app we used the padding function a lot and fixed the position of our containers on to the screen. We didn't foresee this to be a problem until one of us injected the APK into our phone and realized a bunch of things were out of place or misplaced. Continuing on we tried to make our containers more flexible but couldn't do it efficiently without recoding a bunch of things. So, this was one of our biggest issues that we came across. This was by no means a software failure, but more so a failure on our part as we should have known better. Now before one of you guys fall asleep, I'll now let John talk a little bit about our GitHub layout and our final thoughts.

GitHub

Our GitHub has been Separated in different deliverables to screenshot our progress in folders.

Our folder structure is also short to keep readers from getting lost in the branches of our folder tree.

For our first MVP release, we put each of our contributions in different folders as most of us were trying to learn how to code with android studio and flutter we decided to create a section of the app in separate files and merge them to the main source code.

Group reflection

How did you feel about this project? What did you like about it? What did you

dislike?

Overall we feel pretty good. We liked the fast feedback cycle to keep our design evolving and access to more in-depth developer tools too. We didn't like how we couldn't get time with our target customers, the children. And, we would have loved to have more time playing around with our tools more in-depth in order to bring out its full potential, beyond the entry-level utilities.

What did you learn about yourself as you collaborated and worked through this project?

We learned that a team has stark differences in skill levels in different areas. The world changes rapidly and teams need to adapt quickly in order to see a finished project. Different people are good at different things, in order to have a successful MVP, we needed to capitalize on what everybody is good at.

How will you use what you have learned on this project going forward?

One thing we learned is that we should always try and dig deeper with our future customers to help envision the final product.

As we saw in some of our designs we would miss the mark on either priorities or how the interface would be perceived with our target users. Looking into the deeper needs with a customer greatly helps with finalizing our product.