Insight Security
Richard Basdeo
Merna
Eric
MVP Transcript

Slide 1:
Welcome to our Insight Security presentation, and I'm Richard.
I'm Merna.
And I'm Eric! This is our MVP demo presentation.

Slide 2:
Richard: Insight Security is a home security app. It provides users a sense of safety without having to break the bank to afford a home security system. These security systems usually cost around $200 and require a yearly membership. Our app saves you money by turning your old android device into a home security camera, giving purpose to your old device which had none.

We provide security in two ways - the first way is providing a live stream of wherever you place the camera, the second way is via sound by listening to the decibels in the room and notifying the user on another device if there is a noise spike.

Slide 3:
We realized that we would have an issue of screensharing the emulator and running the app at the same time and so instead we decided to record a video of the app running
[Plays video]
So basically you sign into the app with your google account, you pick that you want to be a camera, the decibels will change throughout and once the app detects a noise spike the variable at the top will change to true.

Slide 4: What is left to be accomplished
One thing that I still have to accomplish is making the noise detection more tailored. So right now we have it set so that when it hits 60 decibels,I it will trigger a noise event. But we still want to make it so that the app gathers the average noise level and compares it to that, as opposed to having a one size fits all approach.
[Eric talk about cloud function]
Richard: The last feature that needs to be implemented is the live video streaming. I'm currently working on this with Websockets and realtime streaming protocol, and its the standard for livestreaming today. RTSP basically just controls the streaming services. RTP, which means realtime transport protocol, that controls the transition of the streaming data. I'm currently working on those features and I hope to be done with it in about two weeks or so.

Slide 5:

Richard: Some tools I've used are the google api for logging in and the firebase api for database and authentication.
Merna: And we're currently writing our app in java, because android supports only that and Kotlin (and we decided on Java).
Eric: And I've been using node.js to write cloud functions

Slide 6:
We need to know who was using the app, and one of the easiest ways we found was to use Google Signins to track users signing in. We used the google user signin information to keep track of user accounts.

Slide 7:
On the left side is the google api login screen, the usual login screen that you see with google.

The right side is the authentication page within firebase that shows who was using the app, who was authenticated to use the app. It collects information such as the email, the method used to sign into the app, the date signed in and the identification key. The key allows you to sign in faster so you don't have to repeat the signin process.

Slide 8:
Eric talks about the database

Slide 9:
So basically how the noise event works is that once the user chooses to go into camera mode and designate that phone as a viewer phone, the app starts monitoring the level of noise and once it goes over a certain level a spike is detected. The noise event value is set to true, and there is an image on the next slide which shows the difference

Slide 10:
[Shows the image, talks about the level of noise that will trigger a noise spike]

Slide 11:
So here's where the cloud function comes in -- when the 'noiseEvent' value for that user in the database is set to true, the cloud function sees this and activates. This cloud function sends a notification to each device (users in the database) that have the same username as the device with the noise event. The app then sets 'notified' for each of those users, as I mentioned before, which stops users from being spammed by notifications. Lastly, the cloud function sets noiseEvent back to false so that the cloud function can be called again on later noise events.

Slide 12:
Thank you for watching our presentation!
[open for questions]