

Test Plan for Virtual Front View

Team Team

Prepared by C. Cairns, C. Putinar, M. Morikawa, S. Shojaie, D. Zehavi
For our client Prof. Ghosal

Summary of Product

- Our product is a Vehicle to Vehicle Streaming app designed to run on Android Devices
- It will primarily be a research tool for Professor Ghosal
- This will involve getting two android devices to connect to each other and stream video over a new type of network connection called WiFi-Direct
- Contact Information
 - Matthew Morikawa: Matthew@wcmmr.com 650-465-7511
 - Corina Putinar: cputinar@ucdavis.edu 805-280-2132
 - Cameron Cairns: cgcairns@ucdavis.edu 714-310-6802
 - Shawn Shojaie: sashojaie@ucdavis.edu
 - Doron Zehavi: dzehavi@ucdavis.edu 408-203-7084

Required Resources

Hardware: Two Android Devices with Wifi-Direct and Android 4.0 or better

Software: Virtual Front View Application

We will provide the software, and we will find a way for the group to have access to hardware.

Downloading, Building, and Loading the App

BitBucket Repository is [here](#) and in plaintext below.

<https://bitbucket.org/cputinar/ecs-193a>

There are no actual test scripts for the android app.

In order to build the application and load it onto the phone the android sdk is required, and it helps to install the developer tools. These include the IDE Eclipse.

Install the developer tools and install the SDK [here](#)

<https://developer.android.com/sdk/installing/index.html>

After installing the SDK and developer tools load up the project in eclipse from the repository.

Connect an android device to the computer using a USB.

Compile the project, usually done by pressing the arrow/play button in the Eclipse IDE.

It should prompt you to load it onto an android device.

Functional Test Plan

| User Goal | Test ID | Estimated Time | Estimated Time |
|-------------------------------|---------|----------------|----------------|
| Video Stream Connection | 1 | 1 min | |
| Phone Rotation does not crash | 2 | 1 min | |

Non-Functional Test Plan

| Category | Sub Category | Specific Goal | Use Case Index | Estimated Time (Devs) | Estimated Time (Tester) |
|-----------------|--------------------|---------------------|----------------|-----------------------|-------------------------|
| Fault Tolerance | Distance Tolerance | Clean Shutdown | 3 | 3 min | |
| Fault Tolerance | Server Shutdown | Clean Shutdown | 4 | 1 min | |
| Performance | Time | Lag under 3 seconds | 5 | 1 min | |

List of Test Cases

ID 1

1. Goal: Video Stream Connects
2. Initial Conditions: Both phones turned on with VirtualFrontView Off
3. Required Resources: Two Android Devices and Virtual Front View App
4. Test Script
 - a. Both users launch app
 - b. One user clicks the server button
 - c. Wait till server launches
 - d. Second user clicks client button
 - e. Wait until video stream appears
 - f. Verify that video is being streamed from one user to another

ID 2

1. Goal: Display Rotation Works
2. Initial Conditions: VirtualFrontView App running, and displaying video on both phones
3. Required Resources: Two Android Devices and Virtual Front View App
4. Test Script
 - a. One user rotates phone
 - b. Verify that screen does not rotate, and that the app does not crash

ID 3

1. Goal: Distance Tolerance
2. Initial Conditions: Virtual Front View App running and displaying video on both phones
3. Required Resources: Two Android Devices and Virtual Front View App
4. Test Script
 - a. One user moves away from the other user until the video drops (try 100 meters?)
 - b. Verify that the video goes to home screen

ID 4

1. Goal: Server Shutdown Tolerance
2. Initial Conditions: Virtual Front View App running and displaying video on both phones
3. Required Resources: Two Android Devices and Virtual Front View App
4. Test Script
 - a. User shuts down server application
 - b. User confirms that client application closes stream gracefully

ID 5

1. Goal: Response time on app
2. Initial Conditions: Virtual Front View App running and displaying video on both phones
3. Required Resources: Two Android Devices and Virtual Front View App
4. Test Script
 - a. Server user records easily viewable event - record time
 - b. Client user views easily viewable event - record time
 - c. Confirm that the time difference is less than 3 seconds