Memo

To: Professor Pisano, Professor Osama AlShaykh

From: Allison Durkan, EJ Fitzpatrick, Johan Ospina Buitrago, Santiago Beltran, Tess Gauthier

Team: 13

Date: 2/11/17

Subject: Ventana Second Deliverable Test Plan

1.0 Initiate HoloHub with Windows 10 IoT Core

1.1 Description & Goal

The HoloHub server is the way the internet of things, IoT, devices communicate with the HoloLens running Ventana. Our customer gave us a new constraint that is required us to change the operating system on the raspberry pi, from Raspian, a linux-based OS, to Windows 10 IoT Core. Because of this new constraint, we migrated from Flask to a Node.JS server running on top of the Windows OS. Therefore, we need to test that the raspberry pi starts up properly and reaches stable network connectivity, which is critical for server requests.

1.2 Procedure

Plug Raspberry Pi with the Windows 10 IoT Core SD Card into a 2Amp USB 5V power source. For verification purposes, plug the HDMI cable into the Raspberry Pi and a display.

1.3 Verifiable Result

Once powered on and connected to the network of the router, the ethernet jack status light on the Raspberry Pi will be lit. We will also be able to remote into the pi from the Windows IoT Remote Client application on a computer and see the network connection to “tenda_48e238” from the Win 10 IoT Core settings menu. On the display source of the Raspberry Pi, the IP address should be visible on the desktop.

2.0 Ensure servers operational on HoloHub Node.js Server

2.1 Description & Goal

With our new HoloHub, we needed to find a new solution for controlling sonos speakers. Using our new solution our goal is to test to confirm that all of the same functionality exists through http requests before we test it through the Ventana Application. The HoloHub now runs a Node.JS server. This test will verify if the Node.JS servers are running on the HoloHub.

2.2 Procedure

Utilizing Windows IoT Core Dashboard, open a powershell connection to the HoloHub. Run the commands 'schtasks', record the results of this output. Next, verify the Node executable and server code is stored on the device. On the HoloHub enter:

```
1) cd c:\Node\
   ls
2) cd C:\Projects\HoloHub-js
   ls
3) cd C:\Projects\sonos-http-server
   ls
```

2.3 Verifiable Result

The initial task, `schtasks`, should display TaskNames `HoloHub` and `SonosHTTP` with the status of `Running. Command 1 will return display a Node.exe executable and a corresponding a DLL. Command 2 and 3 will display a server.js file is the test is successful.

3.0 Communicate with HoloHub and Sonos via Node.js Server

3.1 Description & Goal

This test will verify connection to the servers (network flow into the HoloHub) and from the servers to the dependencies (network flow out of the HoloHub). On the Raspberry pi, we have tested to see that the HoloHub and Sonos server modules are operational so next we are going to test that we can communicate with these server modules.

3.2 Procedure

To test the connection to the Sonos HTTP server, we will send a request to the specific Sonos port on the raspberry pi via "192.168.0.105:5005/living room/volume/10". This command, which can be sent from PostMan or a web browser, calls the volume function in the Sonos library. To test the connection to the HoloHub Server, we will send a request to the specific HoloHub Server port on the raspberry pi via "192.168.0.105:8081/status". This command, to be sent from either a web browser or PostMan, requests the status of the HoloHub server from that Node.js module. These requests should be GET requests to the server modules.

3.3 Verifiable Result

On success, the Sonos HTTP server will return a JSON success message.

```
{
    "status": "success"
 }
```

On success, the HoloHub server will return a JSON status message. (this content will vary based on the current status of the HoloHub)


```
{
 "album": "The Temptations Sing Smokey",
 "artist": "The Temptations",
 "title": "My Girl",
 "current_transport_state": "PAUSED_PLAYBACK",
 "uri":
"x-sonos-spotify:spotify%3atrack%3a6jWkZvd1URGktyTTwcpPpB?sid=12&flags=8224&
sn=1",
 "playlist_position": 1,
 "duration": 165,
 "position": "00:02:35",
 "metadata": 155,
 "album_art":
"http://192.168.0.101:1400/getaa?s=1&u=x-sonos-spotify%3aspotify%253atrack%253a
6jWkZvd1URGktyTTwcpPpB%3fsid%3d12%26flags%3d8224%26sn%3d1"
}
```


4.0 Recognize Multiple VuMarks

4.1 Description & Goal

First deliverable testing examined the capabilities of Vuforia, and the maximum allowable distance between the HoloLens' user and Internet of Things device with an image target. Since then, Vuforia's image targets have been switched with VuMarks. VuMarks represent customized images that can have up to 9 bytes of data encoded into them. They can be designed for enhanced recognition capability. In addition, Vuforia is implemented at runtime to associate a VuMark with a specific device's controller. This allows for scalability of Ventana as a platform, and customization for each user.

4.2 Procedure

With the HoloLens on and the Ventana application running, look at VuMark 0. After the hologram appears, look at VuMark 1, and then look at VuMark 2.

4.3 Verifiable Result

An interactive holographic display will show up near each VuMark. VuMark 0 should show the controls for the new music player. VuMark 1 should show the controls for old music player, and VuMark 2 should show the controls for the light bulb. By only showing the hologram that corresponds to the VuMark that the user is gazing at, Ventana provides context to the user.

5.0 Examine Positional Tracking using World Coordinates

5.1 Description & Goal

Microsoft provides an open-source HoloToolkit that utilizes built-in features of the HoloLens, such as spatial mapping. Spatial mapping allows the HoloLens to analyze its surroundings and continuously update its environmental understanding, as its sensors gather more data. The HoloLens maintains a coordinate system for each room, which allows a user to place a hologram and then have it persist across sessions of opening and closing an application, using world coordinates. This provides an opportunity to display the holographic controls to the user during the initial deploy of Ventana using Vuforia, and then the user can move the holographic controls to spaces that make sense in the physical world.

5.2 Procedure

Deploy the version of Ventana that supports world coordinates, located on the feature-WorldAnchorAndVuMarkMerge branch on the VentanaIoT Ventana Github repository. After the user gazes at VuMark 0, and sees the holographic controls, the user double taps anywhere on the hologram. A duplicate of the holographic control will appear in the user's world, and the user moves the new hologram by single tapping on it and dragging it to a new location. Once the hologram is in the desired location, the user taps again to anchor it there. The user repeats the previous steps with VuMark 1, in order to place the second holographic control panel in the world. Next, the user closes Ventana, by doing the bloom gesture and selecting "remove" in the top right corner of the Ventana application window. Then, the user opens Ventana.

5.3 Verifiable Result

When the user double taps on the holographic controls associated with VuMark 0 and VuMark 1, a duplicate of the respective holographic controls should appear. Secondly, the user should have the ability to tap to place the spawned holographic controls anywhere in the physical world, completely independent from gazing at their original VuMark. When the user closes and then opens Ventana, the holographic controls should appear in the same location that they last appeared at, in the previous session of Ventana, due to the utilization of World Coordinates.

6.0 Control Internet of Things Device Using Holographic Display

6.1 Description & Goal

We aim to test the integration of all of the separate components of Ventana, including the VuMarks, spatial coordinates, and the new Windows IoT based HoloHub. Ultimately, we would like to see that the user experience with the holograms has been expanded upon since the first deliverable testing. Features such as the highlighting of the buttons when gazed upon, additional volume controls, and audio confirmation after

tapping a button will be examined. Seeing how this improves the user's overall interaction with the holograms and how it can be further improved represent that main goals of this test.

6.2 Procedure

Gaze upon the music controller and airtap each of the respective controls.

6.3 Verifiable Result

The album artwork should match what the sonos controller application shows. The buttons should display a yellow highlight when gazed upon. The buttons should control their respective functions and make a confirmation sound when pressed.