Verification Test Procedures

Project: Geek Goggles
Team Echo: Hayden Seivewright, Joshua Akinmoluwa
Date: April 17, 2025

1. The device will reflect a rendered display on a micro-display through a collimating lens and prism onto an optical mounted on safety glasses so the user can see an image overlayed a comfortable distance in front of them.

Test	A test image will be displayed, the image is clear and legible when viewed through the device.
Results	*Partial success*
Conclusion	*Device successfully reflects the rendered display onto optical but the image is difficult to clearly see. Proof-of-concept is successful but redesign necessary in the future to improve optics*

2. The device will display various view modes that can be cycled through on the glasses using a simple button UI:
    - Default mode - only displays time for minimal visual invasion
    - Peripheral Display mode:
        i. Peripheral 1 – Displays data from the first connected peripheral (e.g., voltmeter).
        ii. Peripheral X – Displays data from any additional connected peripherals, up to "n" number of peripherals.
    - Document Display mode:
        i. Document 1 – Displays the first uploaded document (e.g., pinout).
        ii. Document X – Displays any additional uploaded documents (e.g., schematics, datasheets), up to “n” number of documents.
    - Camera mode - display view from camera so user can take picture
    - Sensor mode - display basic sensor info, temp, noise, air quality

Test	The device will cycle through each display mode on button click, each mode is clearly displayed. The device is able to cycle back to the original mode
Results	*Success*
Conclusion	*Device successfully cycles through all display modes*

3. The user will be able to photographically document their project with a camera on the device.

Test	The user can view the camera in the camera display mode then take a picture by clicking a button on the device, the picture then successfully uploads to the web app allowing the user to view it.
Results	*Success*
Conclusion	*Device takes a photo on button press and successfully uploads the picture to the server where the user can view it in their project.*

4. The user will be able to document their project by taking voice notes.

Test	The user then selects to record a voice note via button click in basic mode, the note records and uploaded to the web app allowing the user to review the note.
Results	*Success*
Conclusion	*User can record a 10 second voice note the gets uploaded to the server where the user can view it on their project*

5. The Geek Goggles will allow the user to receive alerts on the HUD. A safety alert will set off when the noise quality exceeds 90dBA, or when the air quality exceeds a general ppm of 35 µg/m^3.

Test	Using a speaker and a phone based decibel meter we will create a sound above 90dBA. The device displays a safety alert in the HUD telling the user the environment has a potentially dangerous noise level and to use hearing protection. Then we will create some air contamination from solder smoke and place the device directly above. The device displays a safety alert in the HUD telling the user the environment has potentially dangerous air quality and to move to better ventilation.
Results	*Partial success*
Conclusion	*DB alert successfully triggers at pre-set DB level (Used 60dBA for test as 90dBA is hard to reach). Air quality alert triggers at pre-set AQI (air quality index) level which can roughly correlate to a general ppm but is not as precise.*

6. The device will allow each user to connect the device to a phone or peripheral, with a transfer delay of at most 100ms between the device and the peripheral.

Test	Create a packet of test data from the peripheral with a time-stamp, once the data is received on the device, create another time-stamp and compare the difference between the times
Results	*Partial success*
Conclusion	*Peripheral sends reading data to the device via MQTT with a transfer delay between 150ms-250ms. Early on in the start-up of the device or in heavy network traffic latency can increase.*

7. A user interface on a phone in the form of a web application where the user can upload documents, set timers and view notes.

Test	A pinout PDF will be uploaded on the app and it is viewable on the device HUD in document mode. A 30 second timer will be set in the app and after 30 seconds an alert will be displayed on the device HUD stating the timer is over. A note will be created on the device and can be reviewed on the app.
Results	*Partial Success*
Conclusion	*Documents and notes are successfully uploaded to the device.Voice notes can be created on device are then transcribed to the web app. Timer was not implemented due to time constraints.*

8. The glasses will be attached comfortably to the user to support the weight of the device.

Test	The device will be attached to a willing test subject who will lean over a table simulating a working environment and the device will not fall off the user. The user can hold that position comfortable for 30 seconds. The position held by the user will be a position that they could normally hold for 30 seconds with no issues.
Results	*Success*
Conclusion	*Device is securely attached to the user with a head strap and does not fall off while leaning or moving around*

9. Optionally, depending on available time and project resources we would like to allow the user to use voice commands to change display modes.

Test	While wearing the devices the user will say a set of commands and the device will change to the next display mode.
Results	*Success*
Conclusion	*Voice commands achieve all operations that buttons can, such as changing modes, taking pictures or moving to the next display item.*

10. Optionally, an Augmented reality overlay will be implemented so the user can see details such as voltage while looking at a specific peripheral or schematic based info when looking at a part of the circuit.

Test	The user will directly view a running peripheral and see in the HUD that peripherals data appear.
Results	*Not Implemented*
Conclusion	*Not Implemented due to time constraints*