|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hardware Circuit Testing Sample** | **Software Testing Sample** | **Communication Testing Sample** | **Machine Learning Testing Sample** | **Mobile App Feature Testing Sample** |
| |  |  | | --- | --- | | **Test Name:** | PCB Continuity Testing | | **Performed By:** | Mohammed Ghazal | | **Time/Date** | 15th of March – 4:00pm | | **Test Description:** | In this test, we use the multimeter’s continuity test to ensure current flow in PCB connections without shorts or open circuits | | **Steps:** | 1. Place the design of our PCB next to the fabricated one. 2. Build a table of all connections starting and ending points. 3. Turn the knob on the multimeter to select the continuity test. 4. Connect the leads between the starting and ending points in each connection. 5. Listen for the continuous sound. | | **Expected Results:** | All PCB connections are correct with no short or open circuits | | **Observed Results:** | Connection AF is loose. Consider re-soldering or new PCB. | | **Acceptance Criteria:** | **Fail** | | **Test Result:** | |  |  |  |  | | --- | --- | --- | --- | | **Connection** | **Starting Pt** | **Ending Pt** | **Result** | | PD0 to In1 | PD0 | In1 | Pass | | PD1 to In2 | PD1 | In2 | Pass | | B7 | B | 7 | Pass | | AF | A | F | Fail | | … |  |  |  | | … |  |  |  | |  |  |  |  | | | |  |  | | --- | --- | | **Test Name:** | Testing the digital filter implementation code on Arduino | | **Performed By:** | Mohammed Ghazal | | **Time/Date** | 15th of March – 4:00pm | | **Test Description:** | The purpose of the test is to validate the digital filter designed to process the signal from the sensor. In the test, the output from the sensor x[n] is displayed on the screen. Also, the output of the filter is displayed, i.e., y[n]. If the code works, y[n] should be showing less variations than x[n] because h[n] is a low-pass filter. The code is doing convolution between x[n] and h[n] to produce y[n] using a for loop and if-statements to avoid errors and crashes. | | **Steps:** | 1. Populate h[n] with the filter coefficients. 2. Populate x[n] with dummy data. 3. Produce y[n] by convolving x[n] and h[n]. 4. Print y[n] values. 5. Perform the convolution between h[n] and x[n] on Matlab to produce y[n] 6. Compare y[n] from Matlab from the conv() command with the y[n] calculated using our Arduino code. | | **Expected Results:** | y[n] calculated using our code should match the y[n] produced by Matlab with | | **Observed Results:** | Values calculated from the Arduino implementation are sometimes 5% different from the ones in Matlab. | | **Acceptance Criteria:** | less than 1% error | | **Test Result:** | **Failed.** Must debug and check the code and the math. | | |  |  | | --- | --- | | **Test Name:** | Testing the communication between the Arduino node and the mobile application | | **Performed By:** | Mohammed Ghazal | | **Time/Date** | 15th of March – 4:00pm | | **Test Description:** | The purpose of the test is verifying that the communication protocol messages are delivered to and from the Arduino and the mobile application. | | **Steps:** | 1. Verify that on trigger the signal leaves the Arduino by monitoring the serial communication pins using the oscilloscope. 2. Test the Wifi connection between the Arduino and the router. 3. Send a message from the Arduino to the server and test if the database receives the message. 4. Test that the message is routed and displayed on the mobile application. 5. Verify that on trigger a message leaves the mobile app. 6. Ensure the message content reaches the server and is stored in the database. 7. Check if the message from the server reaches the Arduino | | **Expected Results:** | Two-way communication between the Arduino and the mobile is tested. Latency is calculated and within acceptance criteria. | | **Observed Results:** | * Message from Arduino is received on the mobile app within in 0.53s. * Message from the mobile is received on the Arduino in 2.3s | | **Acceptance Criteria:** | * Message from Arduino is received on the mobile app in less than 1s of latency. * Message from the mobile is received on the Arduino in less than 1s of latency. | | **Test Result:** | **Failed due to latency from mobile to Arduino** | | |  |  | | --- | --- | | **Test Name:** | Testing the machine learning model performance | | **Performed By:** | Mohammed Ghazal | | **Time/Date** | 15th of March – 4:00pm | | **Test Description:** | In this test, we apply K-fold cross validation to evaluate the machine learning model on a limited data sample. | | **Steps:** | 1. Shuffle the dataset randomly. 2. Split the dataset into k groups of approximately equal size. 3. Take one group as a test data set. 4. Take the remaining groups as a training data set. 5. Fit a model on the training set and evaluate it on the test set. 6. Retain the evaluation score and discard the model 7. Summarize the performance of the model using the sample of model evaluation scores. | | **Expected Results:** | The average of our k recorded accuracies is called the average testing accuracy which should be the performance metric of our model. | | **Observed Results:** | Our model doesn’t perform equally on all folds of test sets (average testing accuracy is low), then we’re likely overfitting. | | **Acceptance Criteria:** | * The difference between one-fold accuracy and another shouldn’t exceed 10%. * Accuracy > 90% * Precision > 90% * Recall > 90% | | **Test Result:** | Our model doesn’t generalize well from our training data to unseen data. We need to collect more balanced data or use regularization. | | |  |  | | --- | --- | | **Test Name:** | Testing the functionality of the mobile application | | **Performed By:** | Mohammed Ghazal | | **Time/Date** | 15th of March – 4:00pm | | **Test Description:** | The purpose of this test is to test the geolocation recording feature of the mobile app. | | **Steps:** | 1. Press on the record geolocation data button on the UI interface 2. Move outdoors for 1km 3. Observe pins added on the map 4. Observe database to see locations are logged correctly | | **Expected Results:** | All locations are recorded leading to successful tracking | | **Observed Results:** | Locations are recorded at a frequency of 3 recordings per second. | | **Acceptance Criteria:** | Locations are recorded at a frequency of 5 recording per second. All locations recorded on the phone make it to the database. | | **Test Result:** | **Pass. Works as expected.** | |