5 Testing

5.1 Unit Testing

There is only one unit being tested here: getting the car to run a track. However, there will be multiple algorithms that we come up with to test on this one unit. Additionally, running this unit multiple times is how we teach the robot to pass it.

5.2 Interface Testing

We have 2 main interfaces: the mechanical side of the robot and the software side. The mechanical side will need to be tested for sensor calibration out of the box, bot speed consistency, ability to drive in a straight line when commanded, and movement thresholds. These will be tested as soon as the bot is available to us and will be done with the software provided by AWS for controlling the robot outside of a ML algorithm as well as with conventional testing methods such as finding deviation on a line for straightness testing and using a meter stick to measure its distance over a set time.

For the software interface, we will use the ML simulation to test our robot's functions provided we calibrated the bot correctly it should translate over 1-to-1 and provide a smooth integration of both interfaces. As there are always going to be discrepancies between simulation and real-world performance, we will rely on our calibrations being correct enough for the machine learning algorithm to compensate for this.

A third interface we will encounter is interfacing our design with students and a classroom by creating labs and a club to allow EcpE students the opportunity to learn more about machine learning. This interface testing will be done by working directly with Dr. Rover and Dr. Jones as well as using our own experiences with the project to ensure students will get the best possible learning experience and come out of it with an appreciation for ML.

5.3 Integration Testing

There is only one unit that we are creating in this project which is the machine learning algorithm to reward and train the DeepRacer bot to run a track according to a racing strategy chosen by the team. Like previously mentioned, we will be designing and testing multiple algorithms which we will then integrate together based on how they performed during tested and if they achieved the desired outcome. The integration of these algorithms will be tested the same way as the unit testing, using the DeepRacer simulation and training software. We will also be making sure that the algorithm still works with added sensors to the DeepRacer bot by making sure the bot can still complete multiple laps around a track.

We also want to integrate the DeepRacer into CprE courses. In order to test ways in which the DeepRacer could be included in specific CprE courses, we will be creating small lab assignments using the DeepRacer and machine learning topics that will be reviewed by professors and/or students.

5.4 System Testing

All the tests will involve testing the system as a whole. This is because we require many trials/tests to be ran in order to teach the robot how to function. In this setting of testing, we'll have to ensure that the system we are documenting and presenting will be easily picked up by undergraduate students – if the system, for whatever reason, is unable to be easily delved into – then we will have to look at the system as a whole and append changes as we progress through the project.

In this portion of the testing, we'll incorporate focus groups within introduction to embedded systems and additional minor focus groups on campus – this will provide us the feedback needed to ensure that the educational aspect is on-par with other materials here at the university.

5.5 Regression Testing

As to train the robot it is required to run in a simulation, regression testing will be quite simple as we will only need to implement a new function and restimulate. This will allow us to quickly and easily see if a new addition to our code makes performance worse or interacts poorly with the older code and gives us a good measure for how to compensate for unexpected behavior. After we simulate, we will put the code onto the physical bot and see if it interacts with the new additions in the same way and if not, we will be able to look at the differences between the simulated and physical bot and pinpoint the issue.

5.6 Acceptance Testing

A fully functioning DeepRacer bot will be able to complete multiple laps around any track without going out of bounds. The integration of the DeepRacer bot and CprE courses will be accepted if our designed labs are accepted by professors and students as engaging and beneficial to the material being taught.

5.7 Results

While we specifically cannot provide tables for this portion of the document, as our project isn't spotted only in the engineering field – we will firstly have to approach this complicated embedded systems design and simplify aspects of it, with the intention of replication and education. With this in mind, there are two main goals:

1. The first main goal is for our Deepracer bot to act in a predictable manner, and to perform with the intention of competing in the AWS Deepracer League.
2. The other goal is for our Deepracer bot to act as a platform for undergraduates to approach and learn more about AI, Machine Learning, and so forth – as there truly aren't very many experiences in which the average undergraduate can participate and learn about AI.