**BUCKET INDUSTRIES**

**Acceptance Testing**

\*Grant Ryden

Andrew Dorsaneo

Russell Johnson

Rochelle Lauer

Michael Lee

Nathan Lee

Scott Linkletter

Maria Elena Villamil

# Table of Contents

[Test #1 3](#_Toc415389155)

[Test #2 3](#_Toc415389156)

[Test #3 4](#_Toc415389157)

[Test #4 4](#_Toc415389158)

[Test #5 5](#_Toc415389159)

[Test #6 6](#_Toc415389160)

[Test #7 6](#_Toc415389161)

[Test #8 7](#_Toc415389162)

[Test #9 8](#_Toc415389163)

[Test #10 8](#_Toc415389164)

[Test #11 9](#_Toc415389165)

[Test #12 10](#_Toc415389166)

[Test #13 10](#_Toc415389167)

[Test #14 11](#_Toc415389168)

[Test #15 11](#_Toc415389169)

[Test #16 12](#_Toc415389170)

[Test #17 12](#_Toc415389171)

[Test #18 13](#_Toc415389172)

[Test #19 14](#_Toc415389173)

[Test #20 14](#_Toc415389174)

[Test #21 15](#_Toc415389175)

[Test #22 15](#_Toc415389176)

[Test #23 16](#_Toc415389177)

[Test #24 16](#_Toc415389178)

[Test #25 17](#_Toc415389179)

[Test #26 18](#_Toc415389180)

[Test #27 18](#_Toc415389181)

[Test #28 19](#_Toc415389182)

[Test #29 19](#_Toc415389183)

[Test #31 20](#_Toc415389184)

[Test #32 21](#_Toc415389185)

[Test #33 21](#_Toc415389186)

Acceptance Tests

# Test #1

**Requirements Mapping:** 4

**Tester:**

**Test Date:**

**Pre-conditions:** Have the code for the iPad application available.

**Test**:

Step 1: See that the code is written using Objective C.

**Expected Results:**

The code is written in Objective C.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #2

**Requirements Mapping:** 5

**Tester:**

**Test Date:**

**Pre-conditions:** Have the code for the Arduino available.

**Test:**

Step 1: See that the code is written using C Arduino.

**Expected Results:**

The code is written in C Arduino.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #3

**Requirements Mapping:** 1, 2, 3

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the Bouncing Robot readily accessible.

**Test:**

Step 1: See that there is an Arduino Due in the robot.

Step 2: See that there is a Read Bear Lab Bluetooth Low Energy Shield attached to the Arduino.

**Expected Results:**

There should be an Arduino on the robot for computing and processing. The Bluetooth LE Shield allows the robot to communicate with the iPad application.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #4

**Requirements Mapping:** 6

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the Bouncing Robot readily accessible.

**Test:**

Step 1: Flip the switch to turn on the robot.

Step 2: See that the robot is turned on.

Step 3: Turn off the robot.

**Expected Results:**

The robot should turn on and off from the available switch.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #5

**Requirements Mapping:** 1

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the iPad and Bouncing Robot readily accessible. Test #4 should be passed before testing this case.

**Test:**

Step 1: Turn on the Bouncing Robot.

Step 2: Go to the iPad’s settings and turn on the Bluetooth.

Step 3: Start the application.

**Expected Results:**

The status of the robot should say “Connected”.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #6

**Requirements Mapping**: 9

**Tester:**

**Test Date:**

**Pre-conditions:**

An Arduino running the code for the project must be plugged into a computer running Arduino Studio.

The iOS application must be on.

The serial monitor should be open and set to 9600.

Uncomment the Serial.write that writes the distance received.

**Test:**

Step 1: Click the red button in the upper left corner on the iPad application.

**Expected Results:**

The Arduino should write 150 to the serial log.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #7

**Requirements Mapping:** 7

**Tester:**

**Test Date:**

**Pre-conditions:**

The application must be on, and the robot must be bouncing. Test #6 should be passed before testing this case.

**Test:**

Step 1: Click the red button on the upper left hand corner on the iPad application.

**Expected Results:**

The robot should start slowing down and gradually come to a rest.

The status label should display “Disconnected”.

It should not be possible to enter a new height.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #8

**Requirements Mapping**: 8, 11

**Tester:**

**Test Date:**

**Pre-conditions:**

An Arduino running the code for the project must be plugged into a computer running Arduino Studio.

The iOS application must be on.

The serial monitor should be open and set to 9600.

Uncomment the Serial.Write that writes the distance received.

**Test:**

Step 1: Allow the Bluetooth connection between the Arduino and the application to be established.

Step 2: Using the slider on the application, set the distance the robot should bounce to 20.

Step 3: Check the computer to see that the Arduino writes 20 to the serial log.

**Expected Results:**

The Arduino should write 20 to the serial log.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #9

**Requirements Mapping:** 10

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

The iOS application must be on.

Have XCode running and uncomment the NSLog line of code that prints the string received from the Arduino.

**Test:**

Step 1: Turn the robot on.

Step 2: Turn the application on.

Step 3: Wait until the Bluetooth connection is established.

Step 4: Observe the height and velocity output’s in the application.

**Expected Results:**

The first number should match the robot’s height and the second one should match the robot’s velocity.

**Actual Results:**

**Test Passed / Test Failed**

**Comments**:

# Test #10

**Requirements Mapping:** 12

**Tester:**

**Test Date:**

**Pre-conditions:**

Do not turn the robot on.

Have the iPad with application readily available.

**Test:**

Step 1: Turn the application on.

Step 2: Try to set a height.

**Expected Results:**

When trying to use the slider in the application, a pop up with an error message will appear, stating that the action is invalid until a Bluetooth connection is established.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #11

**Requirements Mapping**: 15

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible.

**Test:**

Step 1: Check the system in order to see whether or not the robot is securely attached to the two vertical linear guides.

**Expected Results:**

The robot should be firmly attached to both of the linear guides.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #12

**Requirements Mapping:** 16

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available, and the microcontroller should be visible.

**Test:**

Step 1: Check whether or not the microcontroller is securely attached to the robot.

Step 2: Verify that the microcontroller is sufficiently surrounded with enough shock protection in order to prevent any damage due to bouncing.

**Expected Results:**

The microcontroller should be firmly attached and sufficiently protected

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #13

**Requirements Mapping:** 17

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: See that the robot’s gearbox material is made of aluminum.

**Expected Results:**

The gearbox of the robot should be aluminum because it is a relatively light material.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #14

**Requirements Mapping:** 18

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: See that the robot’s structure holding the different components material is made of ABS plastic.

**Expected Results:**

The frame of the robot should be 3d printed because it is a relatively light material.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #15

**Requirements Mapping:** 20

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the Mathematica model readily available.

**Test:**

Step 1: Check to see that the masses of the pendulums and body can be manually changed in the model.

Step 2: Input some values into the model.

Step 3: Run the model and see that it can properly run (assuming the input values work).

**Expected Results:**

The Mathematica model should be able to be easily modified for testing the system before it can be built up.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #16

**Requirements Mapping:** 21

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the Mathematica model readily available.

**Test:**

Step 1: Check to see that the spring stiffness can be manually changed in the model.

Step 2: Input a value into the model.

Step 3: Run the model and see that it can properly run (assuming the input value works).

**Expected Results:**

The Mathematica model should be able to be easily modified for testing the system before it can be built up.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #17

**Requirements Mapping:** 22

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the Mathematica model readily available (with the default values provided).

**Test:**

Step 1: Check to see that the model runs properly, the graphic should represent what the system should do.

**Expected Results:**

The Mathematica model is used to help visualize the motion of the system.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #18

**Requirements Mapping**: 24

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test**:

Step 1: See that there are two springs fixed to the bottom of the structure.

**Expected Results:**

The springs should be attached to the structure so that it does not add additional weight to the robot.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #19

**Requirements Mapping**: 26

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: Check to see that the motor is securely fixed to the robot.

Step 2: Have the robot bounce a few times, whether using the application or just manually bouncing the robot.

Step 3: Recheck to see that the motor has not been dislodged or displaced.

**Expected Results:**

The motor should not experience any significant movement such that it is displaced or damaged. If the motor becomes non-operational, then the robot becomes useless.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #20

**Requirements Mapping:** 27

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: Check to see that there is an encoder fixed to the gear assembly.

**Expected Results:**

There should be an encoder to determine the angular position of the pendulums. Only one encoder is necessary as both pendulums are driven by the same motor.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #21

**Requirements Mapping:** 28

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: See that the robot’s components are all properly housed and placed in an orderly fashion.

**Expected Results:**

The structure of the robot should be spacious enough to effectively and efficiently house all of its components.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #22

**Requirements Mapping**: 29

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test**:

Step 1: Check to see that each pendulum is connected to a shaft which connects it to the gearbox assembly.

**Expected Results:**

Each pendulum should be fixed to its own rod which is in turn fixed to the gear assembly. This will allow the pendulums to rotate in opposite directions.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #23

**Requirements Mapping:** 35

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: Check to see that the microcontroller and other components are stacked in a vertical manner.

**Expected Results:**

The main stationary components should be stacked vertically so that the system’s center of mass stays centrally located.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #24

**Requirements Mapping:** 30

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

Test #23 should be satisfied before testing this case.

**Test:**

Step 1: Check if the axis the pendulums move along relatively align with the center of mass of the system.

**Expected Results:**

The two axes of the pendulums and the center of mass of the system should be relatively aligned so as to maximize the vertical motion of the robot.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #25

**Requirements Mapping:** 31, 32

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible.

**Test**:

Step 1: See that there is one motor in the robot.

Step 2: See that it is a DC brushed motor.

**Expected Results:**

There should only be one motor in the robot and it should be a DC brushed motor.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #26

**Requirements Mapping**: 33

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible.

**Test:**

Step 1: See there are ball bearings attached to the shafts attached to the pendulums.

**Expected Results:**

There should be two sets of ball bearings on each shaft that has a pendulum.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #27

**Requirements Mapping:** 34

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available.

**Test:**

Step 1: Check to see that the components are mounted in place using fasteners.

**Expected Results:**

The components should be adequately fit and mounted onto the frame of the robot.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #28

**Requirements Mapping:** Missing

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible on the vertical linear track.

**Test:**

Step 1: Manually spin one of the pendulums.

**Expected Results:**

Moving one pendulum should cause the other pendulum to rotate in the opposite direction.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #29

**Requirements Mapping:** 30

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible on the vertical linear track.

**Test:**

Step 1: Turn the robot on.

Step 2: Fully compress the springs by gently pushing the robot down

Step 3: While firmly holding the robot down with the springs completely compressed, spin the pendulums to make sure there is enough clearance for the pendulums to swing without contacting the base plate.

**Expected Results:**

There will be enough clearance between the robot and the base plate when the robot is at its lowest possible point.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

Test #: 30 I AM QUESTIONABLE

Requirements Mapping: 6

Tester:

Test Date:

Pre-conditions:

Have the robot readily accessible on the vertical linear track.

Test:

Step 1: Turn the robot on.

Step 2: Start the robot in motion.

Step 3: Turn off the robot.

Expected Results:

The system effectively powers off.

Actual Results:

Test Passed / Test Failed

Comments:

# Test #31

**Requirements Mapping:** 38

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible on the vertical linear track.

**Test:**

Step 1: Turn the robot on.

Step 2: Start the robot in motion.

Step 3: Check if the power source is a battery mounted on the body of the robot.

**Expected Results:**

The robot is powered by a battery that is mounted on the body of the robot.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #32

**Requirements Mapping**: 39

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily accessible on the vertical linear track.

**Test:**

Step 1: Measure the structure from the top surface of the base plate to the bottom surface of the top plate.

**Expected Results:**

The linear guides are approximately three feet in height.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**

# Test #33

**Requirements Mapping:** 37

**Tester:**

**Test Date:**

**Pre-conditions:**

Have the robot readily available, all Arduino, iOS application, and robot requirements should be passed.

**Test:**

Step 1: Turn the robot on.

Step 2: Set the robot in motion.

Step 3: Looking at the iPad application, see that the vertical position of the robot is constantly updated.

**Expected Results:**

The robot uses a linear position sensor to determine its location which can then be used to determine other metrics.

**Actual Results:**

**Test Passed / Test Failed**

**Comments:**