ENGINEERING

McMaster University

Project Three – There's A Recyclable Among Us:

Design a System for Sorting and Recycling Containers

ENGINEER 1P13 – Integrated Cornerstone Design Projects

Tutorial 17

Team Fri-35

Buu Ha (hab8)

Joshua Currie (currij15)

Harikashan Thayeswaran (thayeswh)

Muhammad Danyal Afzal (afzalm7)

Submitted: March 7, 2021

Table of Contents

Academic Integrity Statement 4

Project Schedule 6

Initial Gantt Chart 6

Final Gantt Chart 7

Logbook 7

Scheduled Weekly Meeting Agendas 7

Design Studio Worksheets 7

List of Sources 8

Appendix A – Solid Model Screenshots: 9

Final Model 9

Hopper and Cage 10

Stopper for Rungs 11

Rod for Rungs 11

Rod for Linkage 12

Stopper for Linkage Rod 12

Linkage 1 13

Linkage 2 13

Appendix B – Fully-dimensioned Engineering Drawings: 14

Hopper 14

Hopper Cage 15

Rung Rod Stopper 16

Rung Specific 17

Linkage Stopper 18

Linkage Rod 19

Linkage 1 20

Linkage 2 21

Appendix C – Exploded Assembly Drawing: 22

Exploded Assembly Drawing 22

Parts List 23

Exploded Drawing Close-Up 24

..... 24

Appendix D – Output Grapher of Simulation 25

Output graph of angle of the hopper 25

Appendix E – Screenshots of Computer Program: 26

Screenshot of Computer Program 26

Loading Container 29

Q-Arm Placing Container on Hopper 29

Q-Bot Depositing Container 30

Q-Bot Following Loop to Return Home 30

Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Muhammad Danyal Afzal 400307161

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Buu Ha 400264438

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Harikashan Thayeswaran 400326364

Hari.T

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Joshua Currie 400316897

Joshua Currie

Executive Summary

This project was to design a system that sorts and recycles containers of different materials. Two sub-teams were made to handle these tasks, the modelling-team and computation team. The computation team oversaw controlling the dispenser, the q-arm and the q-bot to perform the functions needed. The modelling sub-team oversaw the hopper's creation for the containers, creating a dumping mechanism, and making a simulation of the hopper depositing the containers.

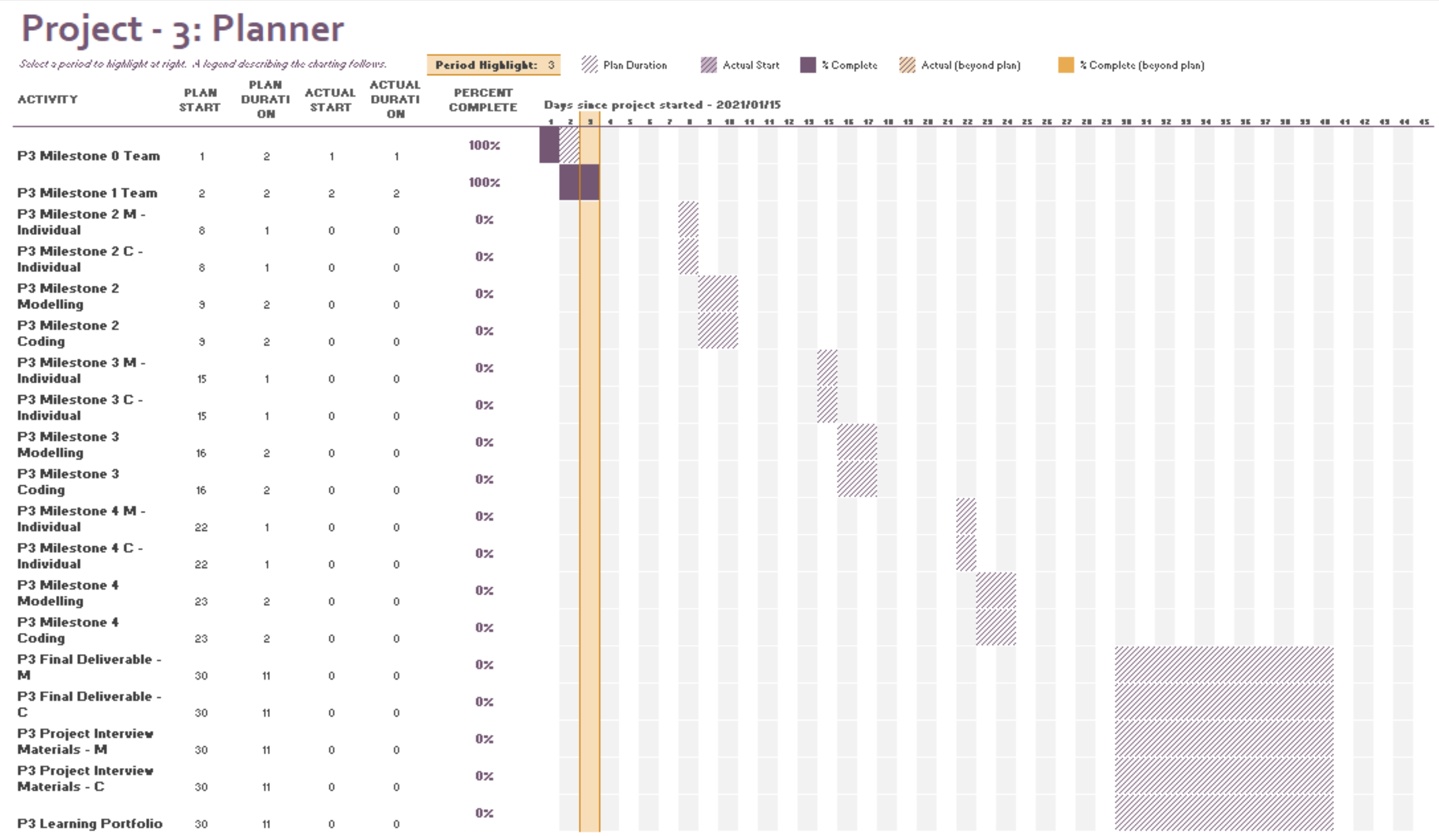
The design solution for the modelling sub-team included a hopper that could contain three containers and a rotating system that allowed the hopper to rotate about an axis to deposit the containers in a bin. Furthermore, a dynamic simulation was completed to ensure the functionality of the design. The hopper features a section of caged wall to keep the containers in the hopper without adding excess material and weight. This design allows less energy needed to rotate the hopper while restricting the containers' movement to ensure it deposits in the bin correctly. The depositing mechanism features a linear actuator that connects to a set of linkages that raises the hopper. As the linear actuator extends, the linkages bend and come into contact with the hopper, which lifts it to the desired position. Two smaller linkages were chosen over one large linkage to promote balance by decentralizing the force acting on the hopper. The contact point on the linkage is rounded to reduce friction on the hopper and allows a smooth elevation of the hopper to deposit the containers. This resulted in a smooth deposit by the computing team, as they were able to translate the data provided by the modelling team to their code.

The computing sub-team's code had to include five functions controlling the environment. First, the computation sub-team had to create a workflow and pseudocode to get an idea of how to make each function work. Code was then created from the pseudocode and workflow while making minor changes along the way. The code dispenses a container and outputs its specific properties (type, mass, and ID). The code then loads that container onto the hopper onto a specific location. The code checks to see if the next container's properties make it go to the same bin to be loaded onto the hopper at the same time to be more efficient. After that process is done, the q-bot (holding the hopper) moves across a yellow line towards the specific bin using the ultrasonic sensor, deposits the container(s) and returns to its initial position to repeat the process. To deposit the container, the modelling sub-team provided a list of rotation angles and times that were accessed in the code to control the actuator and hopper to replicate a dumping process.

Main Body

Project Schedule

Initial Gantt Chart



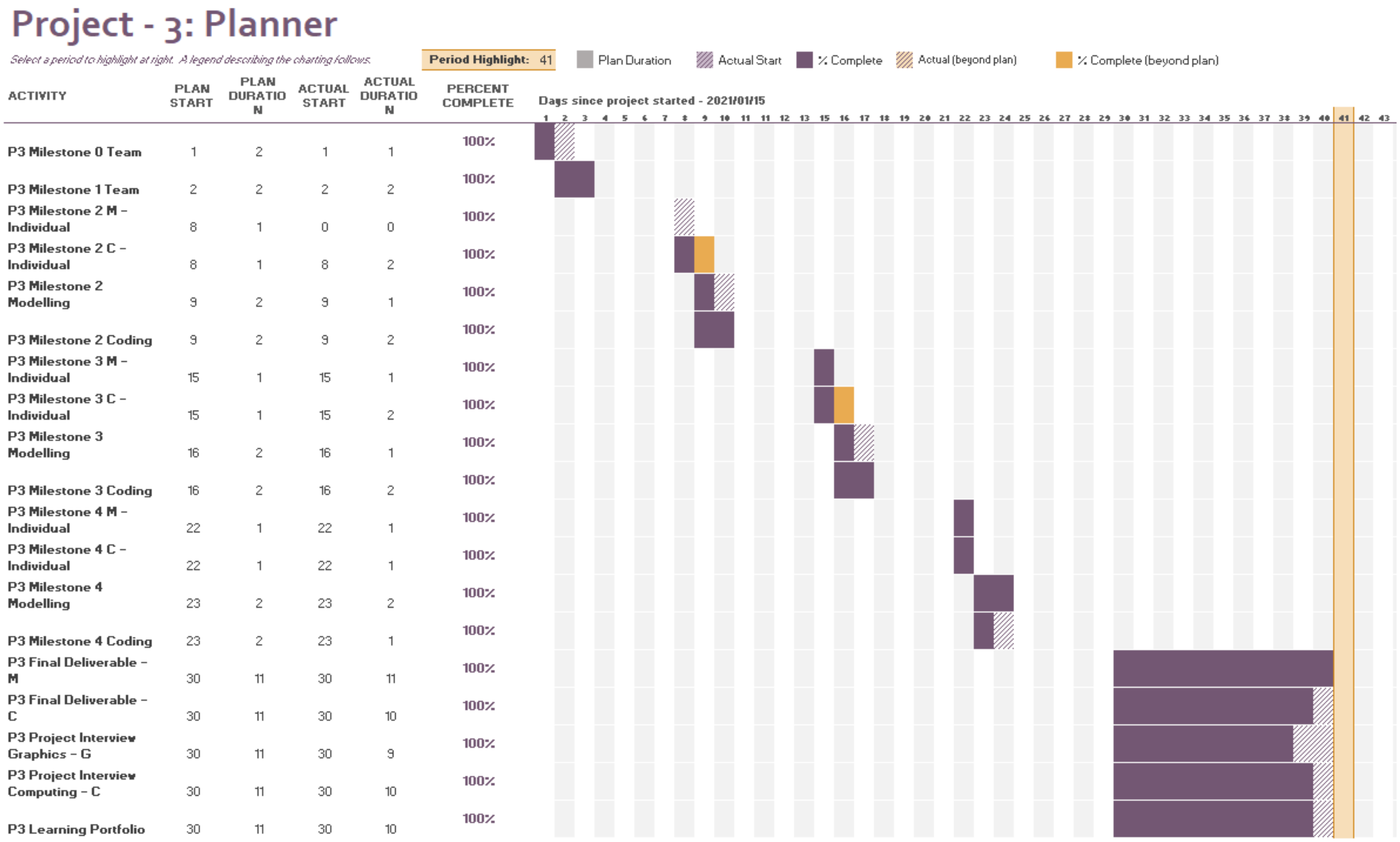
Project - 3: Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 3

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
P3 Milestone 0 Team	1	2	1	1	100%
P3 Milestone 1 Team	2	2	2	2	100%
P3 Milestone 2 M - Individual	8	1	0	0	0%
P3 Milestone 2 C - Individual	8	1	0	0	0%
P3 Milestone 2 Modelling	9	2	0	0	0%
P3 Milestone 2 Coding	9	2	0	0	0%
P3 Milestone 3 M - Individual	15	1	0	0	0%
P3 Milestone 3 C - Individual	15	1	0	0	0%
P3 Milestone 3 Modelling	16	2	0	0	0%
P3 Milestone 3 Coding	16	2	0	0	0%
P3 Milestone 4 M - Individual	22	1	0	0	0%
P3 Milestone 4 C - Individual	22	1	0	0	0%
P3 Milestone 4 Modelling	23	2	0	0	0%
P3 Milestone 4 Coding	23	2	0	0	0%
P3 Final Deliverable - M	30	11	0	0	0%
P3 Final Deliverable - C	30	11	0	0	0%
P3 Project Interview Materials - M	30	11	0	0	0%
P3 Project Interview Materials - C	30	11	0	0	0%
P3 Learning Portfolio	30	11	0	0	0%

Final Gantt Chart



Project - 3: Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 41

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
P3 Milestone 0 Team	1	2	1	1	100%
P3 Milestone 1 Team	2	2	2	2	100%
P3 Milestone 2 M – Individual	8	1	0	0	100%
P3 Milestone 2 C – Individual	8	1	8	2	100%
P3 Milestone 2 Modelling	9	2	9	1	100%
P3 Milestone 2 Coding	9	2	9	2	100%
P3 Milestone 3 M – Individual	15	1	15	1	100%
P3 Milestone 3 C – Individual	15	1	15	2	100%
P3 Milestone 3 Modelling	16	2	16	1	100%
P3 Milestone 3 Coding	16	2	16	2	100%
P3 Milestone 4 M – Individual	22	1	22	1	100%
P3 Milestone 4 C – Individual	22	1	22	1	100%
P3 Milestone 4 Modelling	23	2	23	2	100%
P3 Milestone 4 Coding	23	2	23	1	100%
P3 Final Deliverable – M	30	11	30	11	100%
P3 Final Deliverable – C	30	11	30	10	100%
P3 Project Interview Graphics – G	30	11	30	9	100%
P3 Project Interview Computing – C	30	11	30	10	100%
P3 Learning Portfolio	30	11	30	10	100%

Logbook

Logbook found here:

Scheduled Weekly Meeting Agendas

Milestone 0&1: Agenda and Meeting Minutes here

Milestone 2: Agenda and Meeting Minutes here

Milestone 3: Agenda and Meeting Minutes here

Design Review: Agenda and Meeting Minutes here

Work Period: Agenda and Meeting Minutes here

Design Studio Worksheets

Milestone 0: Worksheet here

Milestone 1: Worksheet here

Milestone 2: Worksheet here

Milestone 3: Worksheet here

Milestone 4: Worksheet here

List of Sources

[1] H. M. M. El-Hageen, “A New Technique for Improving the Estimation of a Reflective Optical Color Sensor,” Sensing and Imaging, vol. 21, no. 1, pp. 1–19, Dec. 2020, doi: 10.1007/s11220-020-0276-5.

[2] “Light Sensor including Photocell and LDR Sensor,” Basic Electronics Tutorials, 15-Feb-2018. [Online]. Available: https://www.electronics-tutorials.ws/io/io_4.html. [Accessed: 19-Jan-2021].

[3] “Photoelectric Sensors,” OMRON. [Online]. Available: https://www.ia.omron.com/support/guide/43/introduction.html [Accessed: 19-Jan-2021].

[4] “What is an ultrasonic / level sensor?,” KEYENCE. [Online]. Available: https://www.keyence.ca/ss/products/sensor/sensorbasics/ultrasonic/info/. [Accessed: 22-Jan-2021].

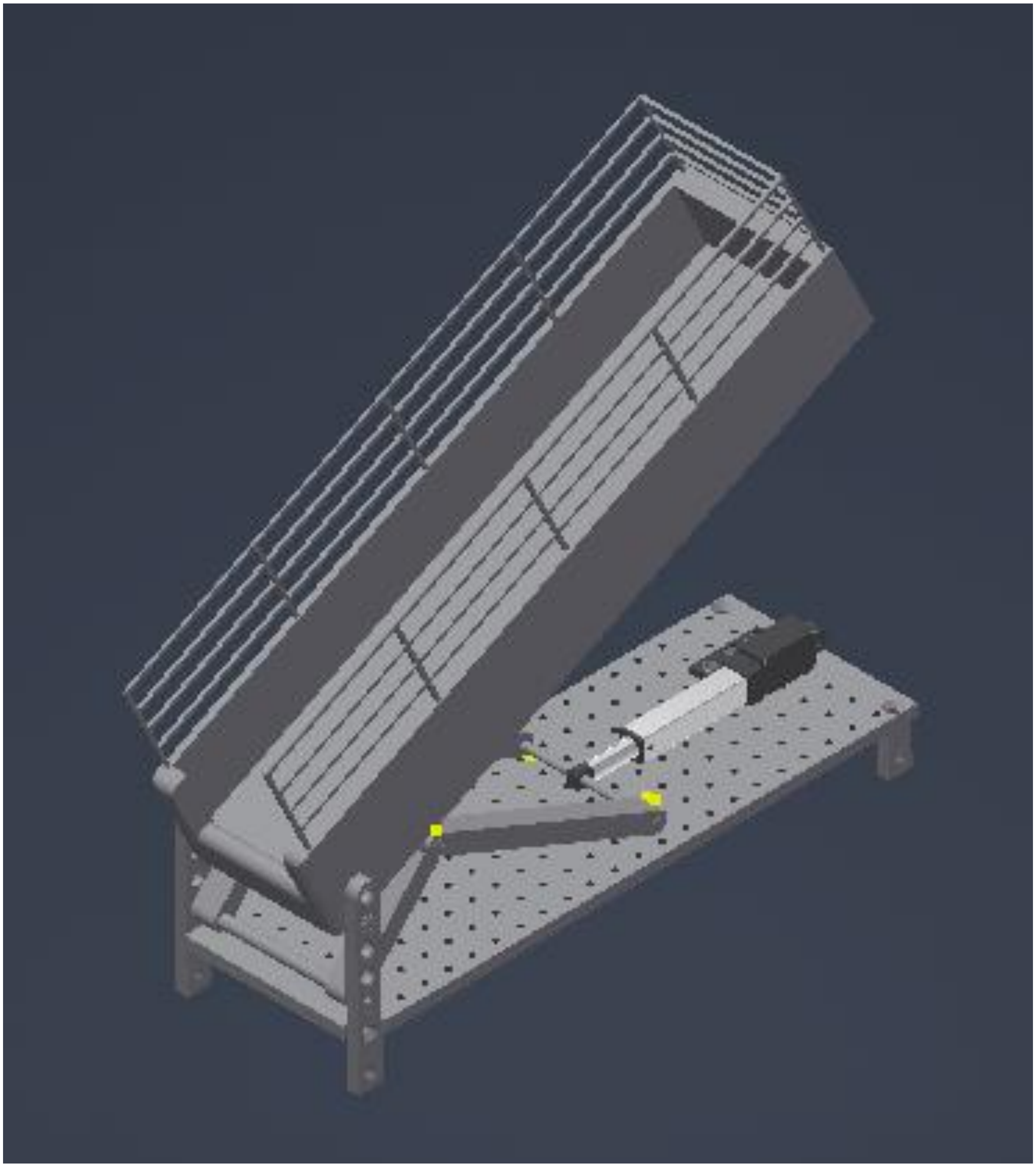
[5] D. Jost, “What is a Hall Effect Sensor?,” FierceElectronics, 08-Oct-2019. [Online]. Available: https://www.fierceelectronics.com/sensors/what-a-hall-effect-sensor. [Accessed: 22-Jan-2021].

[6] Hotron, “Infrared Motion Sensors for Automatic Doors,” Hotron, 26-Nov-2020. [Online]. Available: https://hotron.com/technology/active-infrared-door-sensors/. [Accessed: 22-Jan-2021].

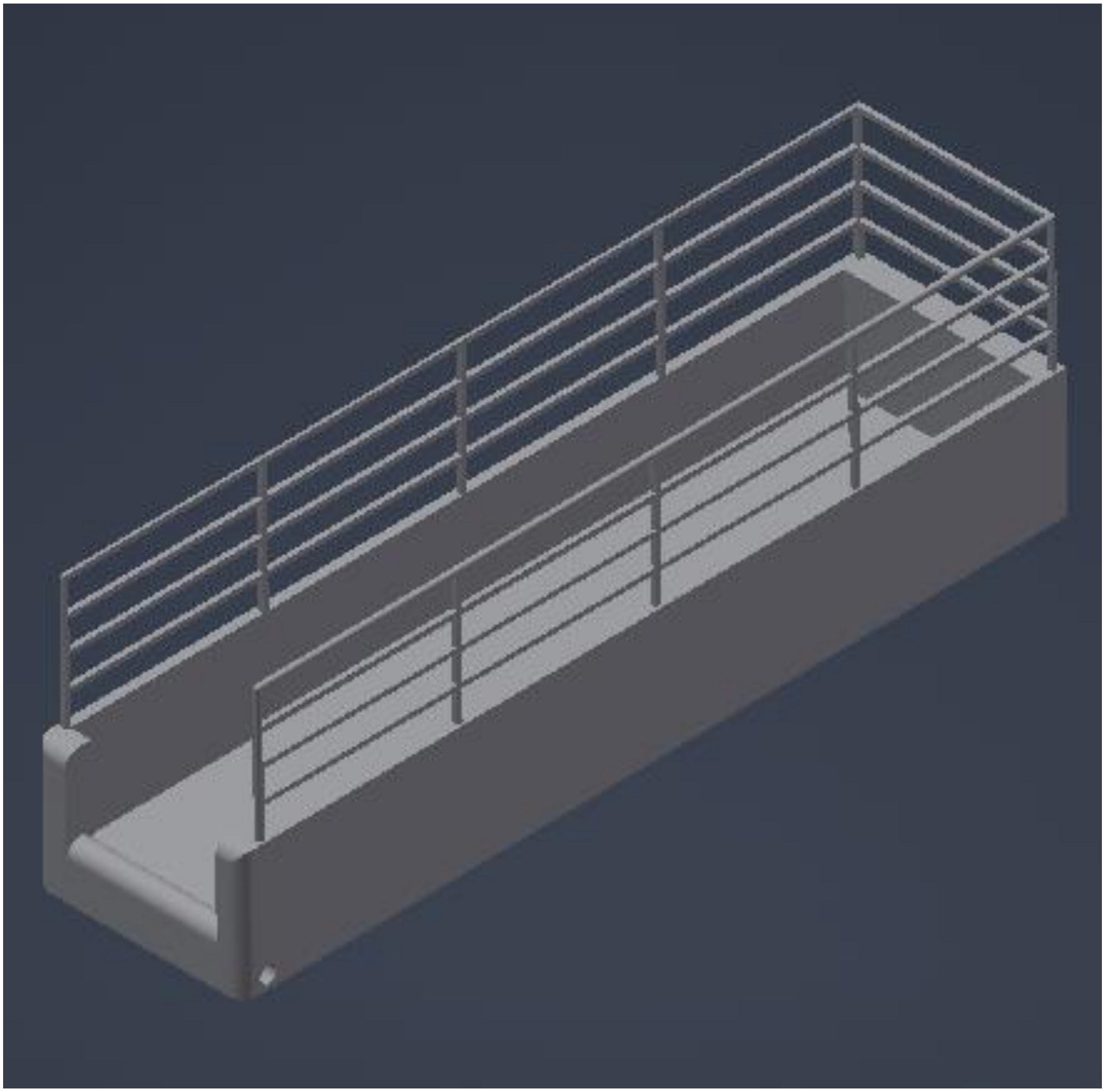
[7] “P3 Python Library Documentation.” McMaster University, Hamilton.

Appendix A – Solid Model Screenshots:

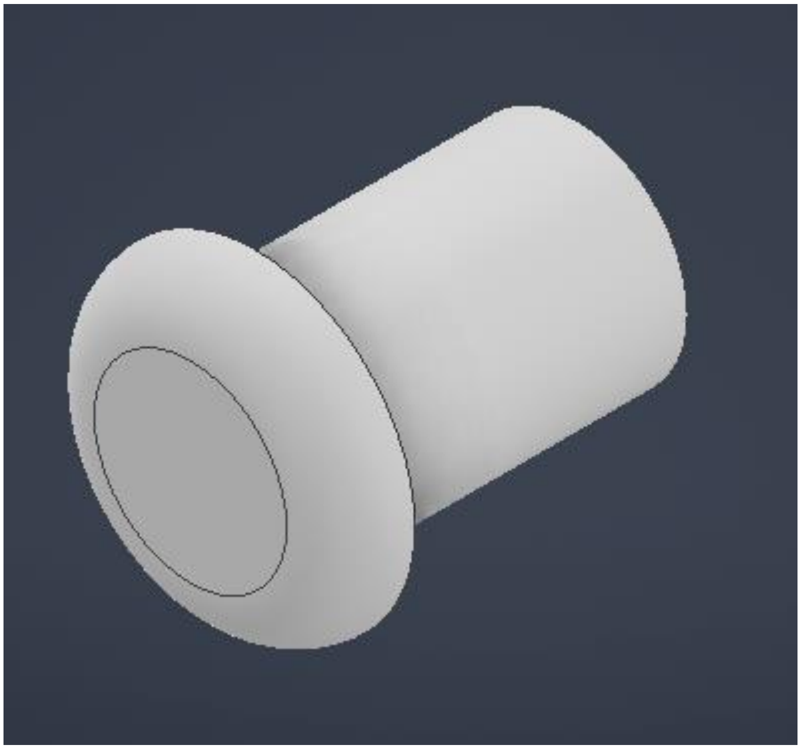
Final Model



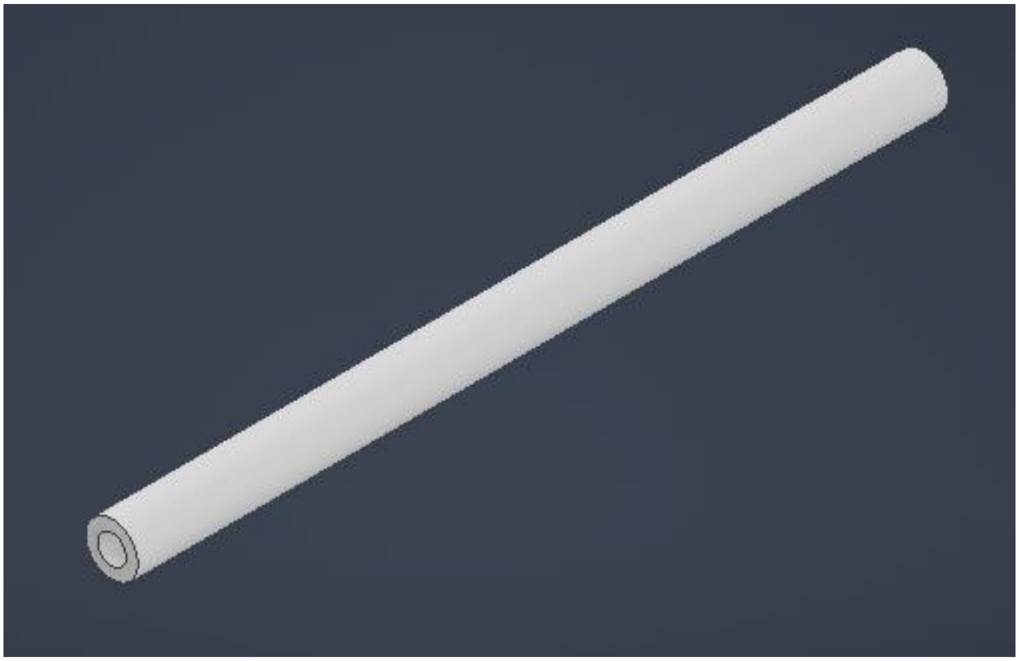
Hopper and Cage



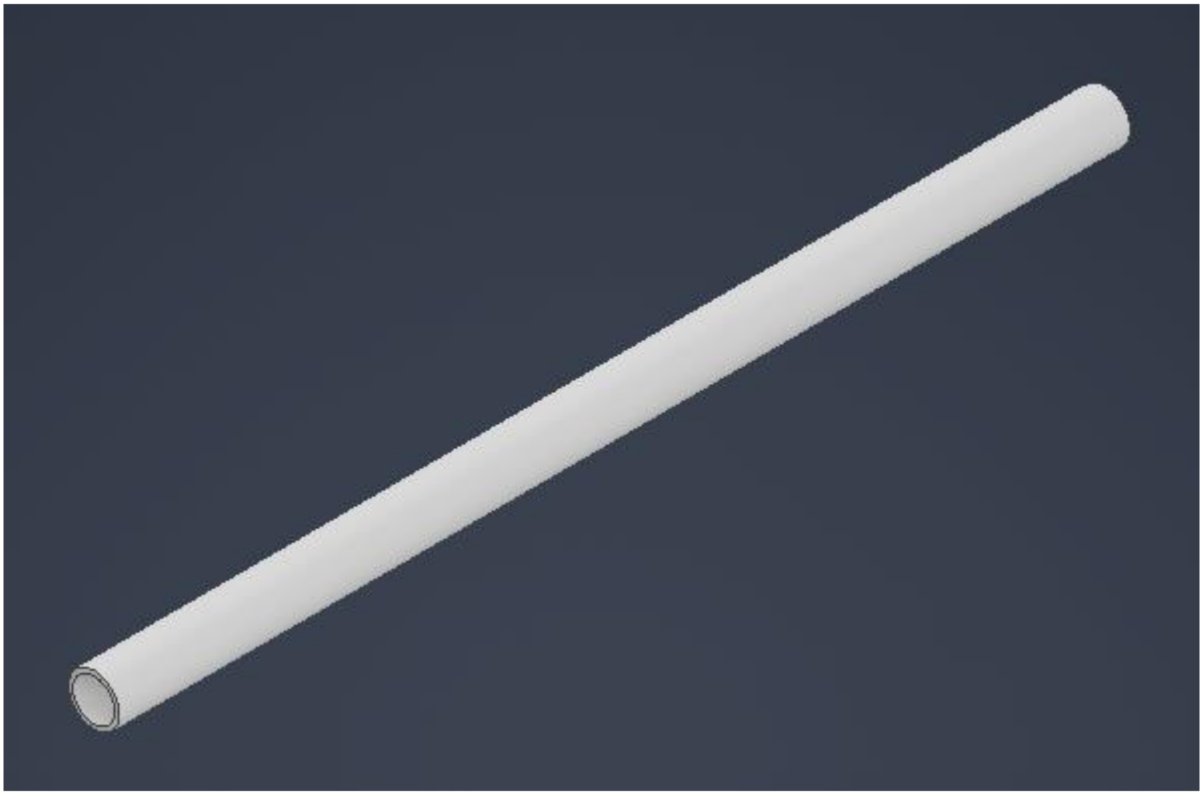
Stopper for Rungs



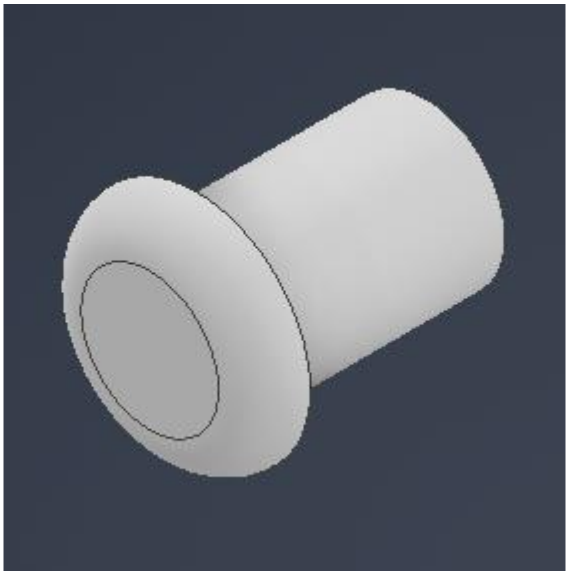
Rod for Rungs



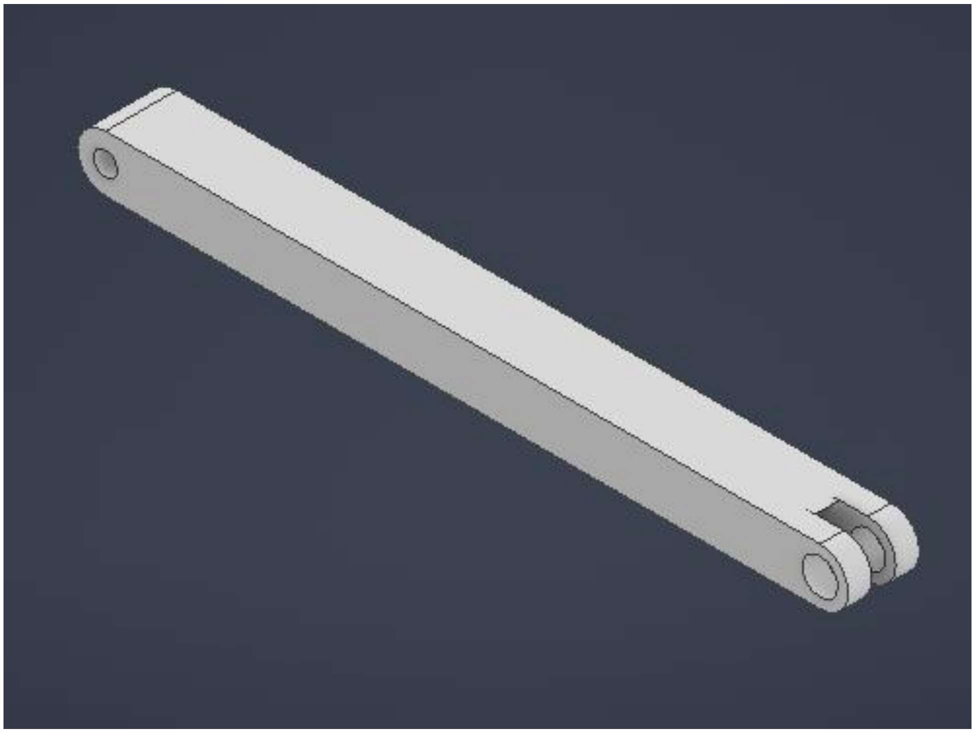
Rod for Linkage



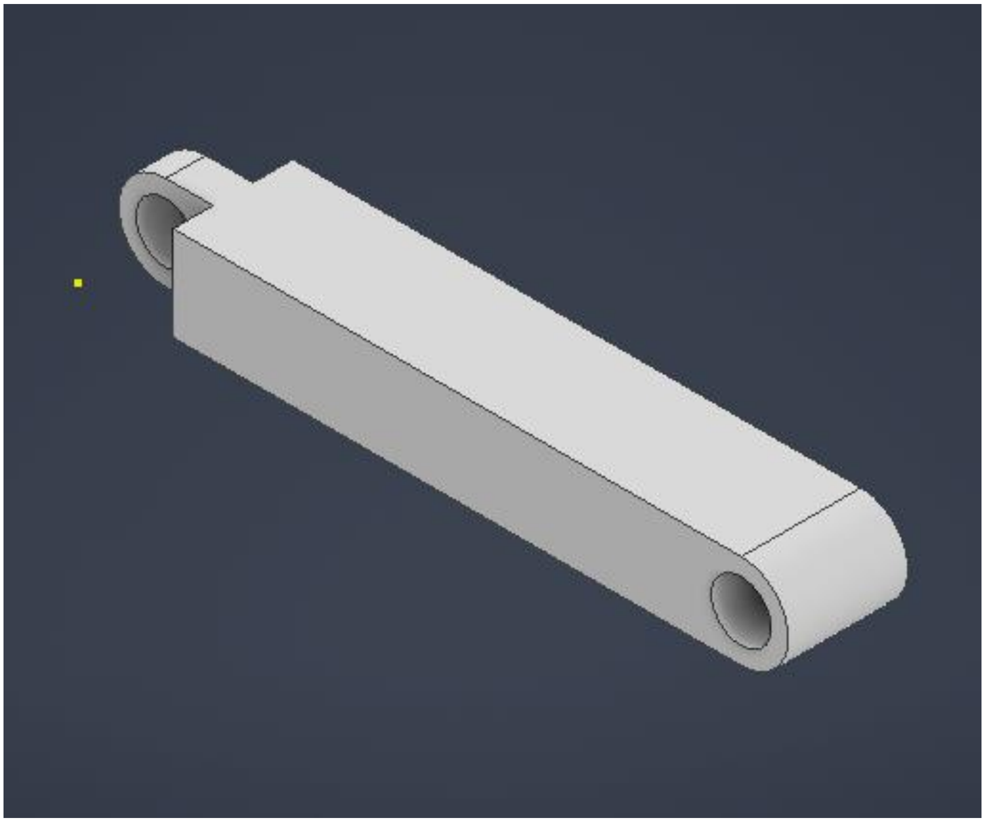
Stopper for Linkage Rod



Linkage 1

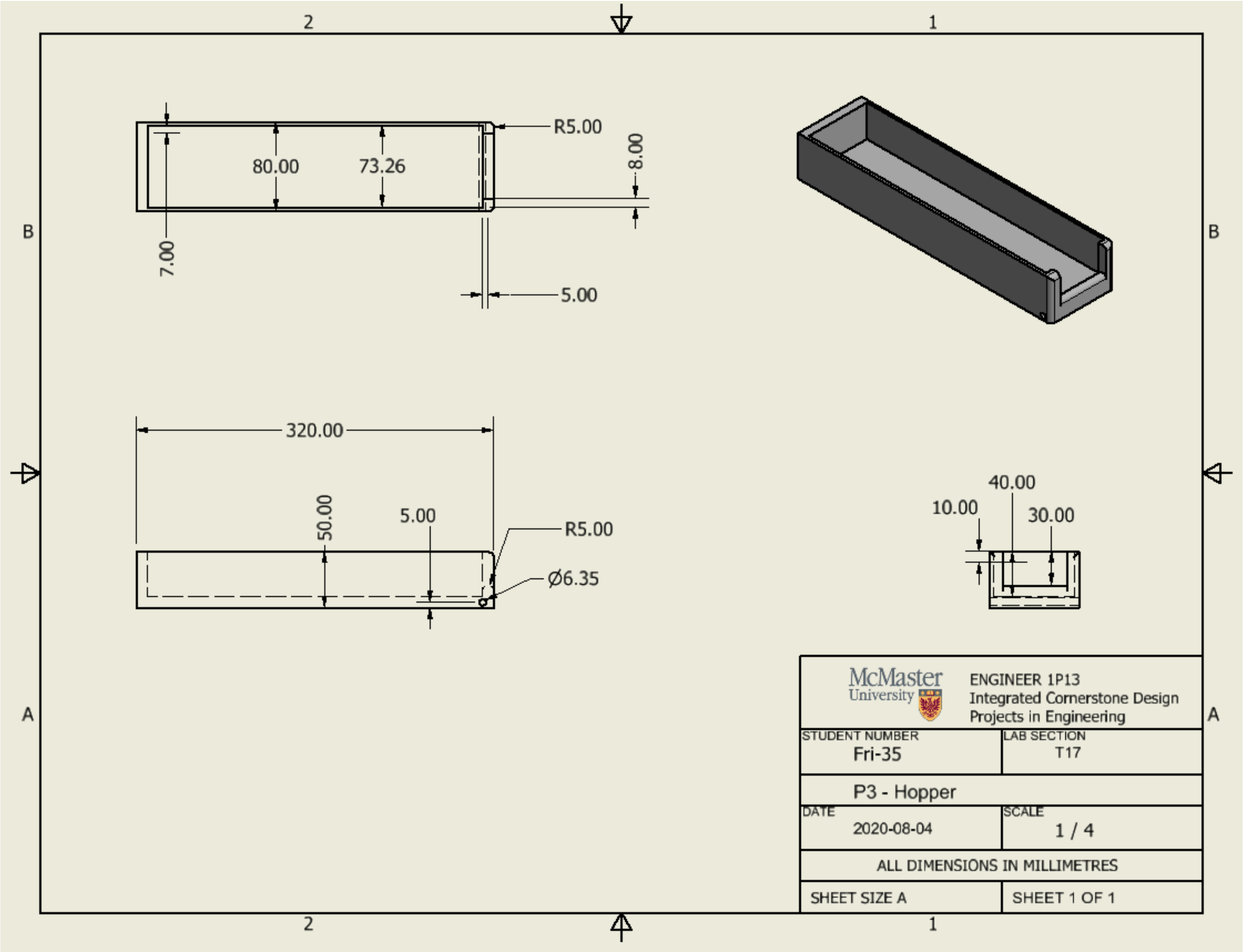


Linkage 2



Appendix B – Fully-dimensioned Engineering Drawings:

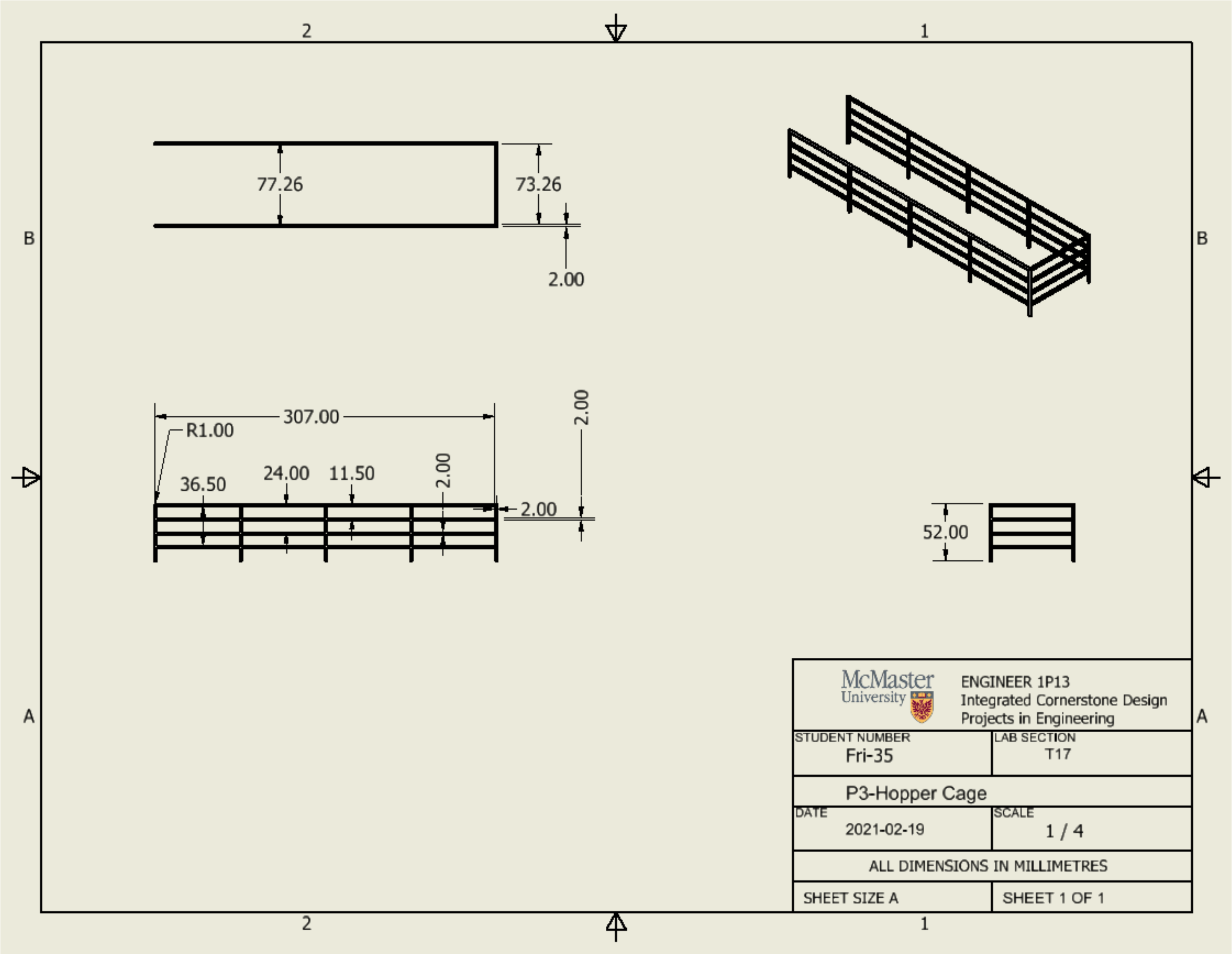
Hopper



ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering

STUDENT NUMBER Fri-35	LAB SECTION T17
P3 - Hopper	
DATE 2020-08-04	SCALE 1 / 4
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

Hopper Cage

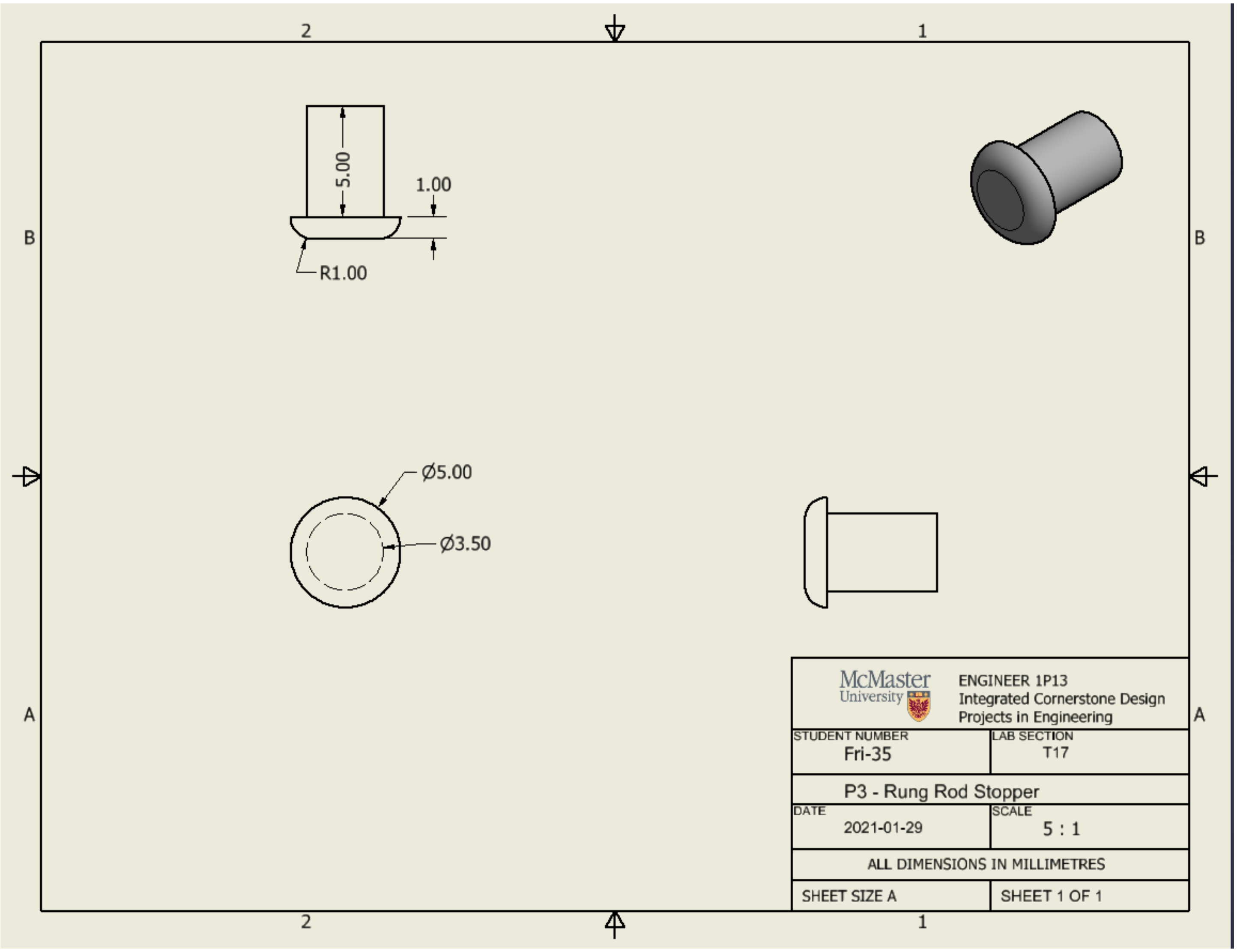


McMaster University

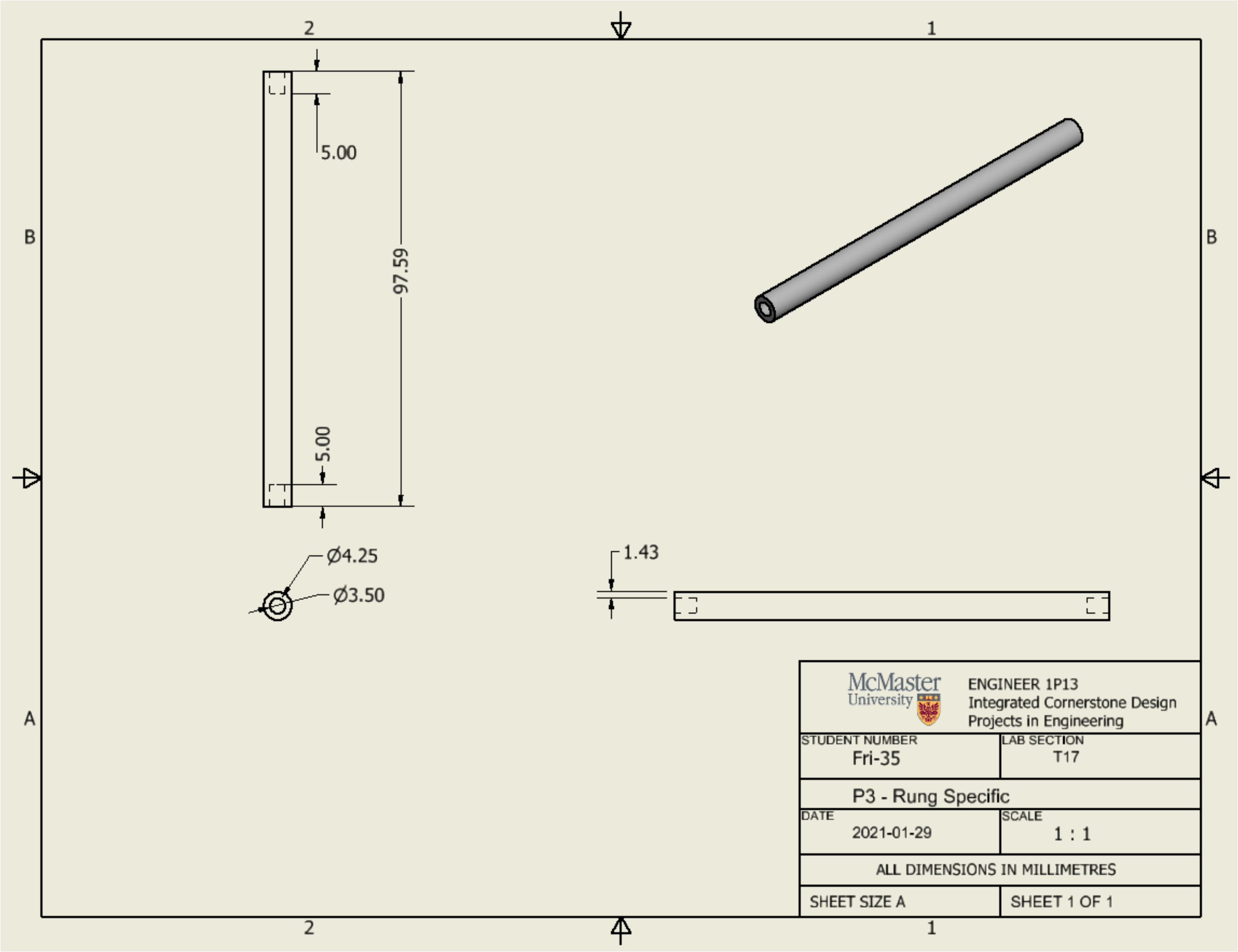
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering

STUDENT NUMBER Fri-35	LAB SECTION T17
P3-Hopper Cage	
DATE 2021-02-19	SCALE 1 / 4
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

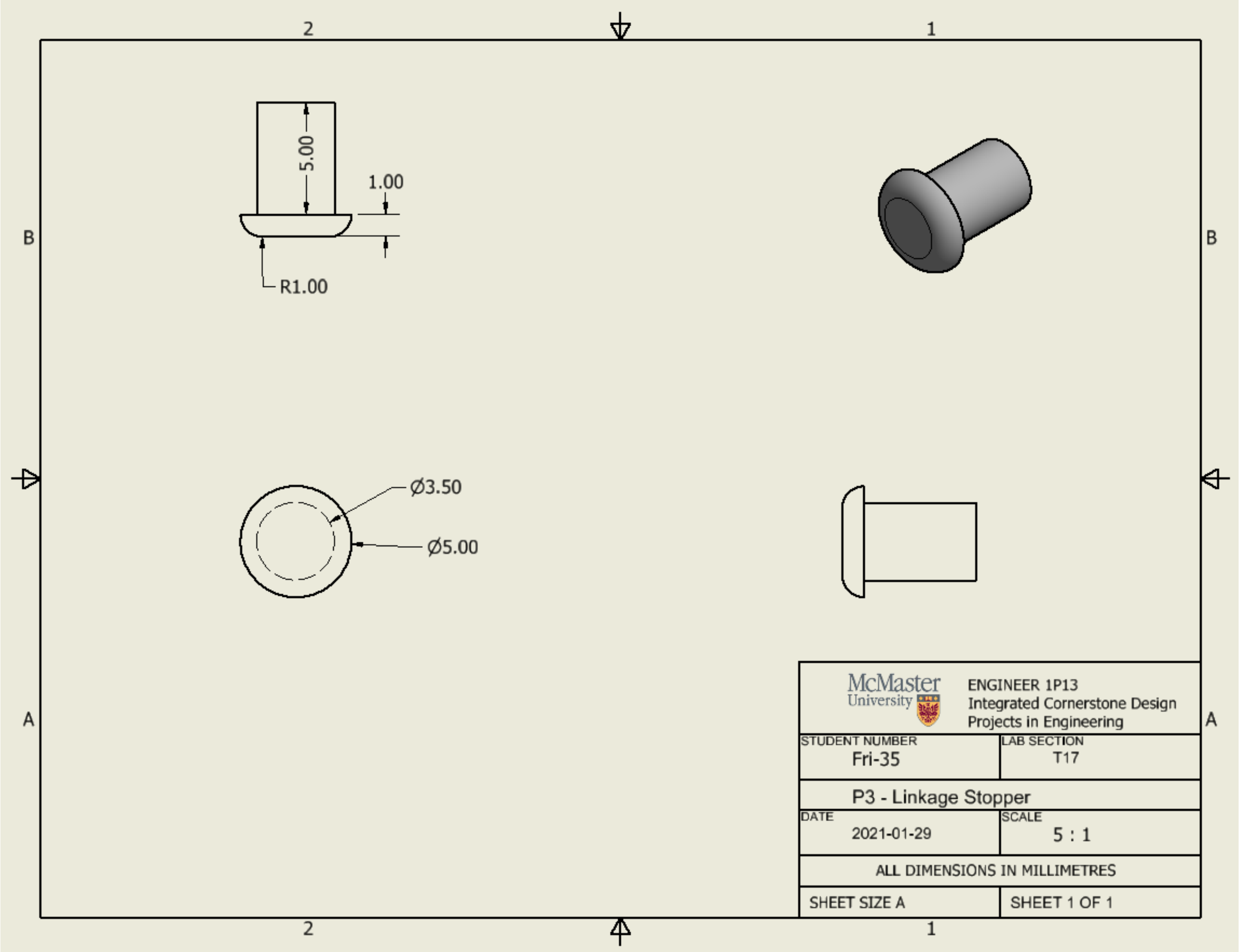
Rung Rod Stopper



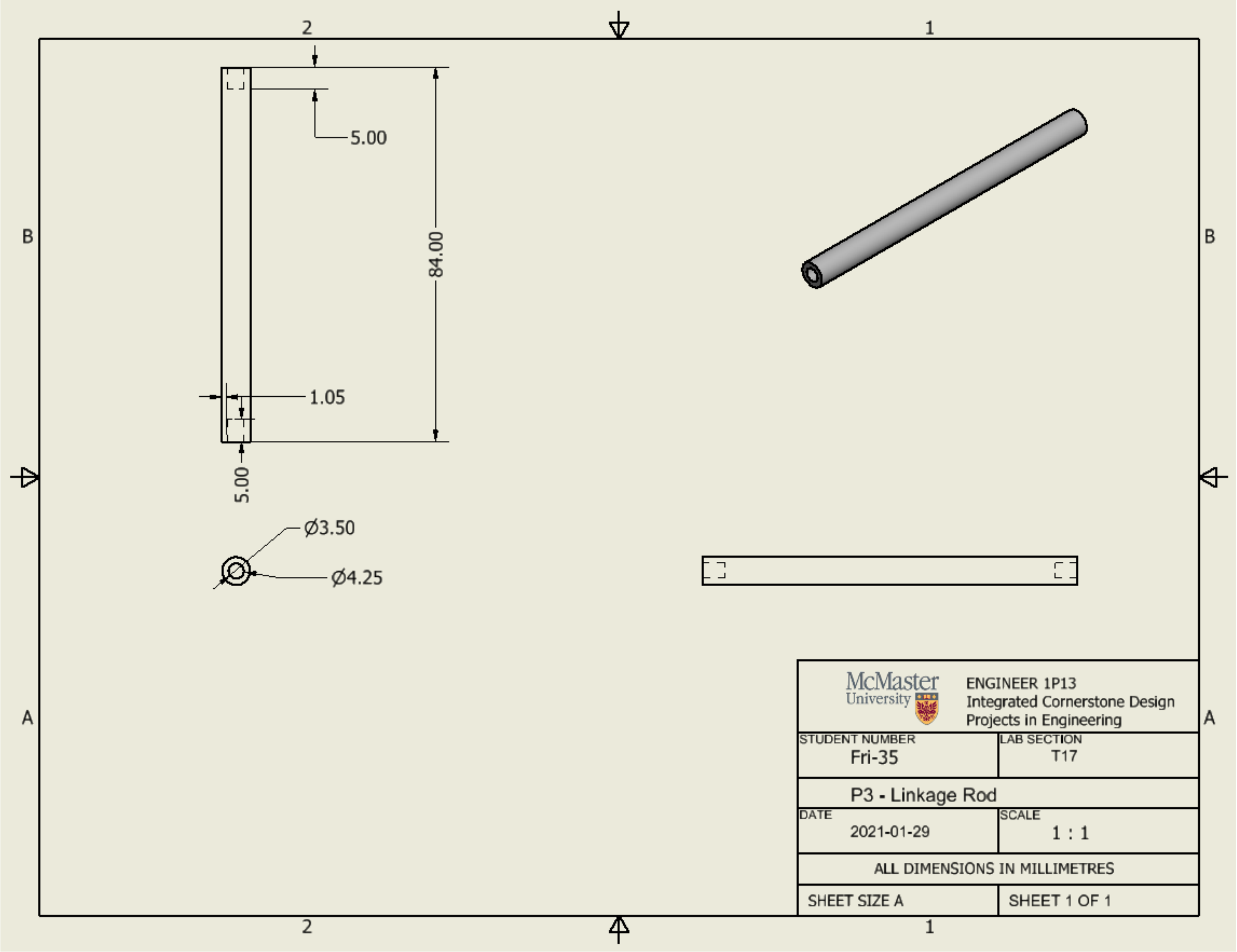
Rung Specific



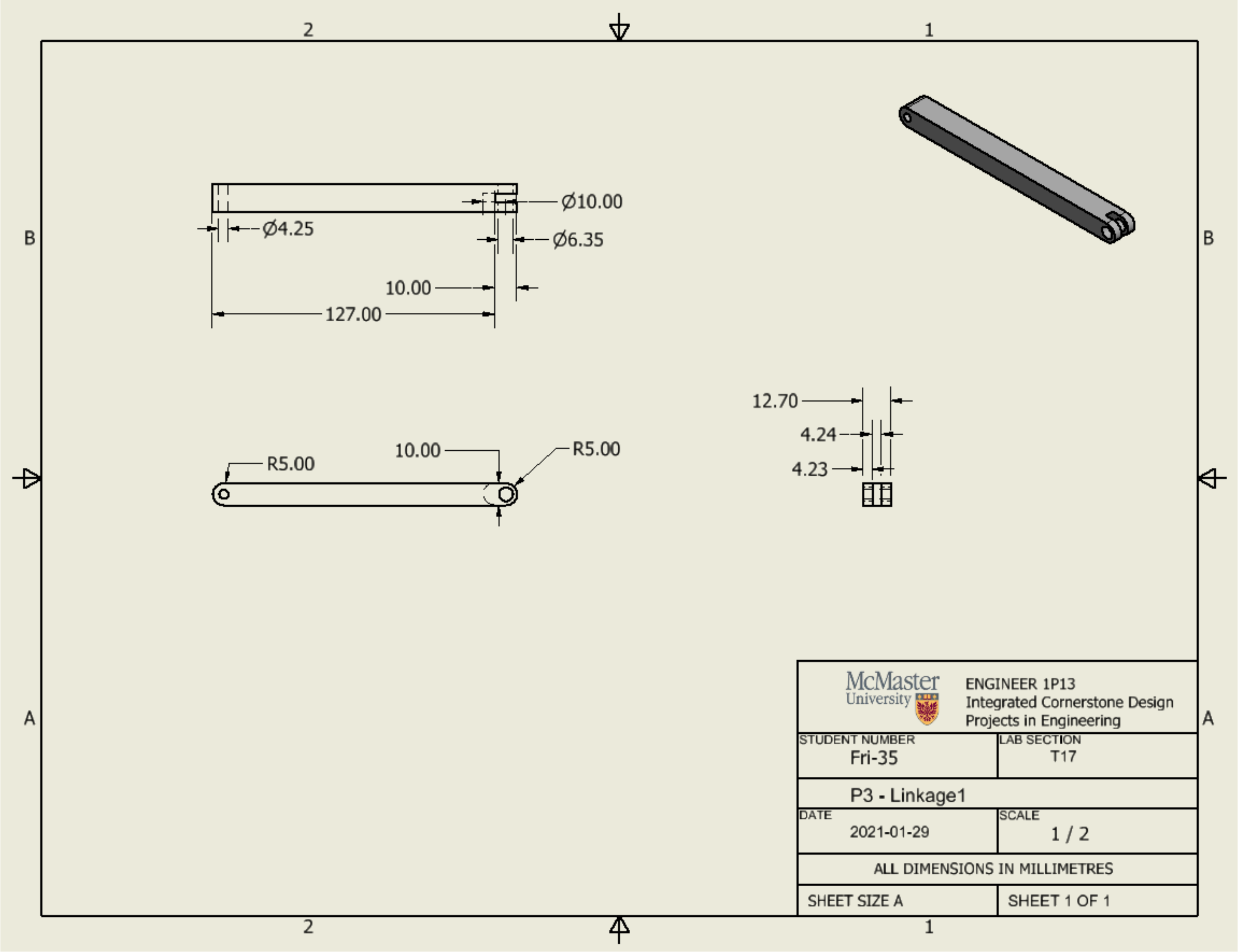
Linkage Stopper



Linkage Rod

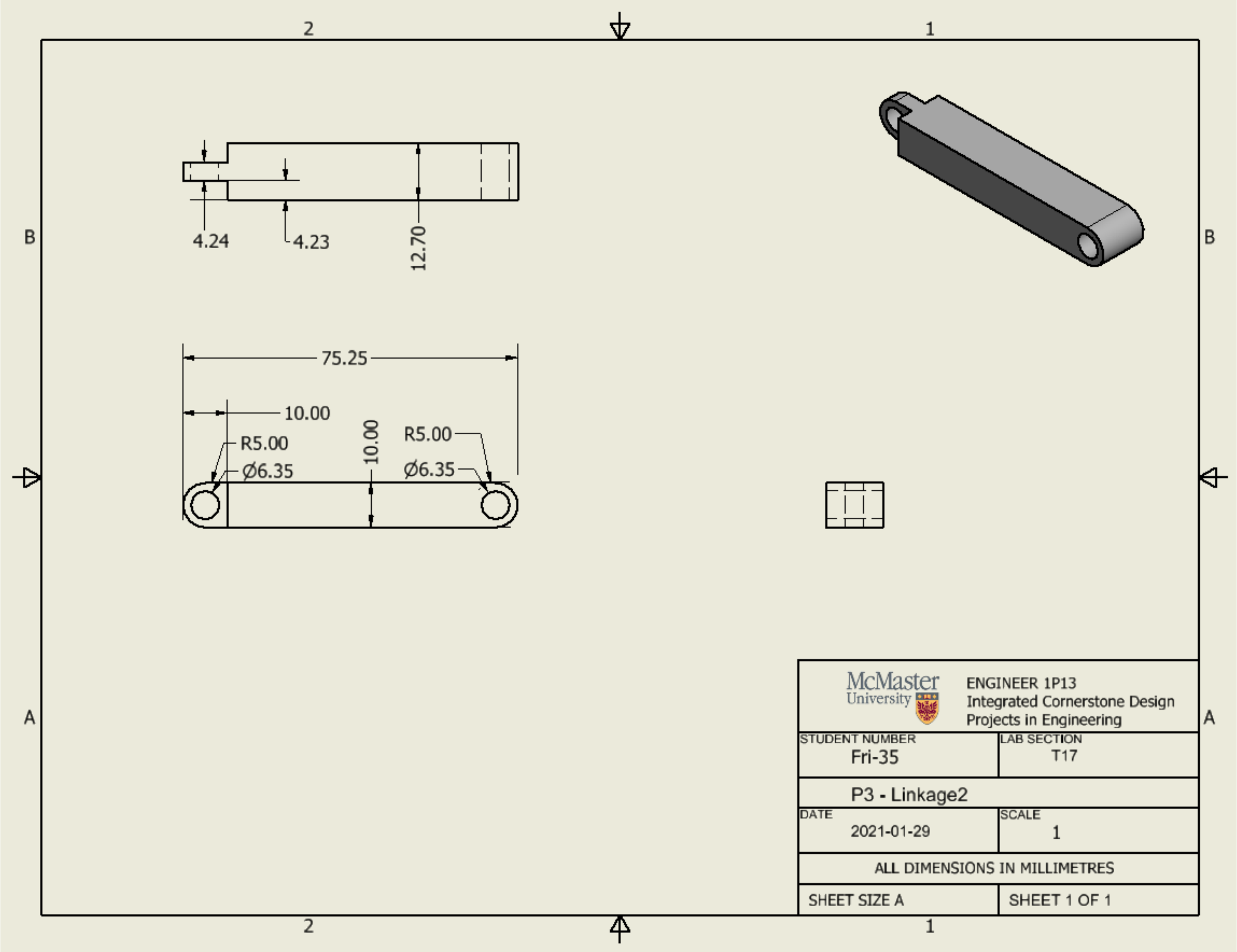

2
1
B
A
5.00
84.00
1.05
5.00
Ø3.50
Ø4.25
McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering
STUDENT NUMBER
Fri-35
LAB SECTION
T17
P3 - Linkage Rod
DATE
2021-01-29
SCALE
1 : 1
ALL DIMENSIONS IN MILLIMETRES
SHEET SIZE A
SHEET 1 OF 1

Linkage 1



ENGINEER 1P13 Integrated Cornerstone Design Projects in Engineering	
STUDENT NUMBER Fri-35	LAB SECTION T17
P3 - Linkage1	
DATE 2021-01-29	SCALE 1 / 2
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

Linkage 2

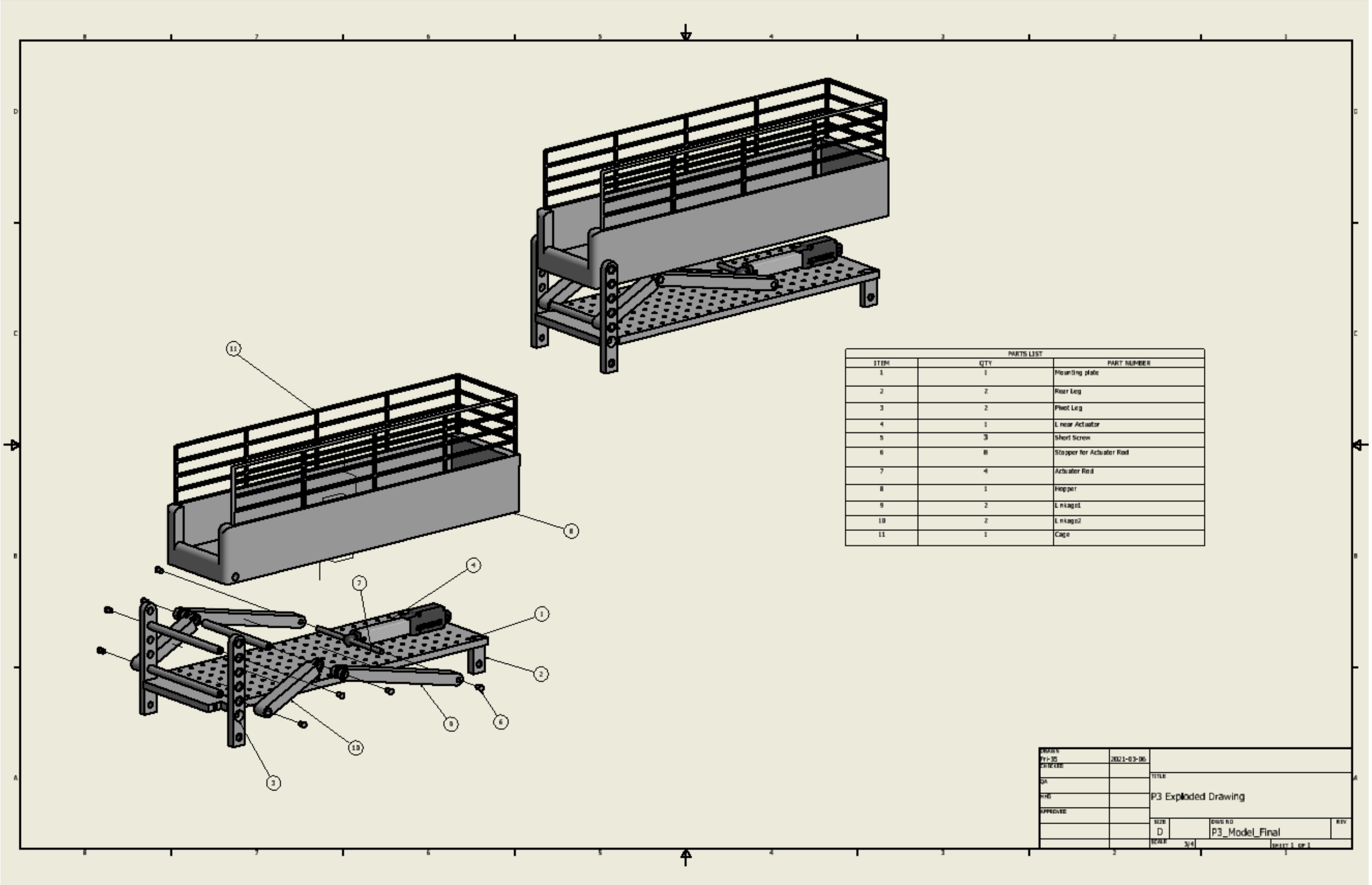


McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering

STUDENT NUMBER Fri-35	LAB SECTION T17
P3 - Linkage2	
DATE 2021-01-29	SCALE 1
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

Appendix C – Exploded Assembly Drawing:

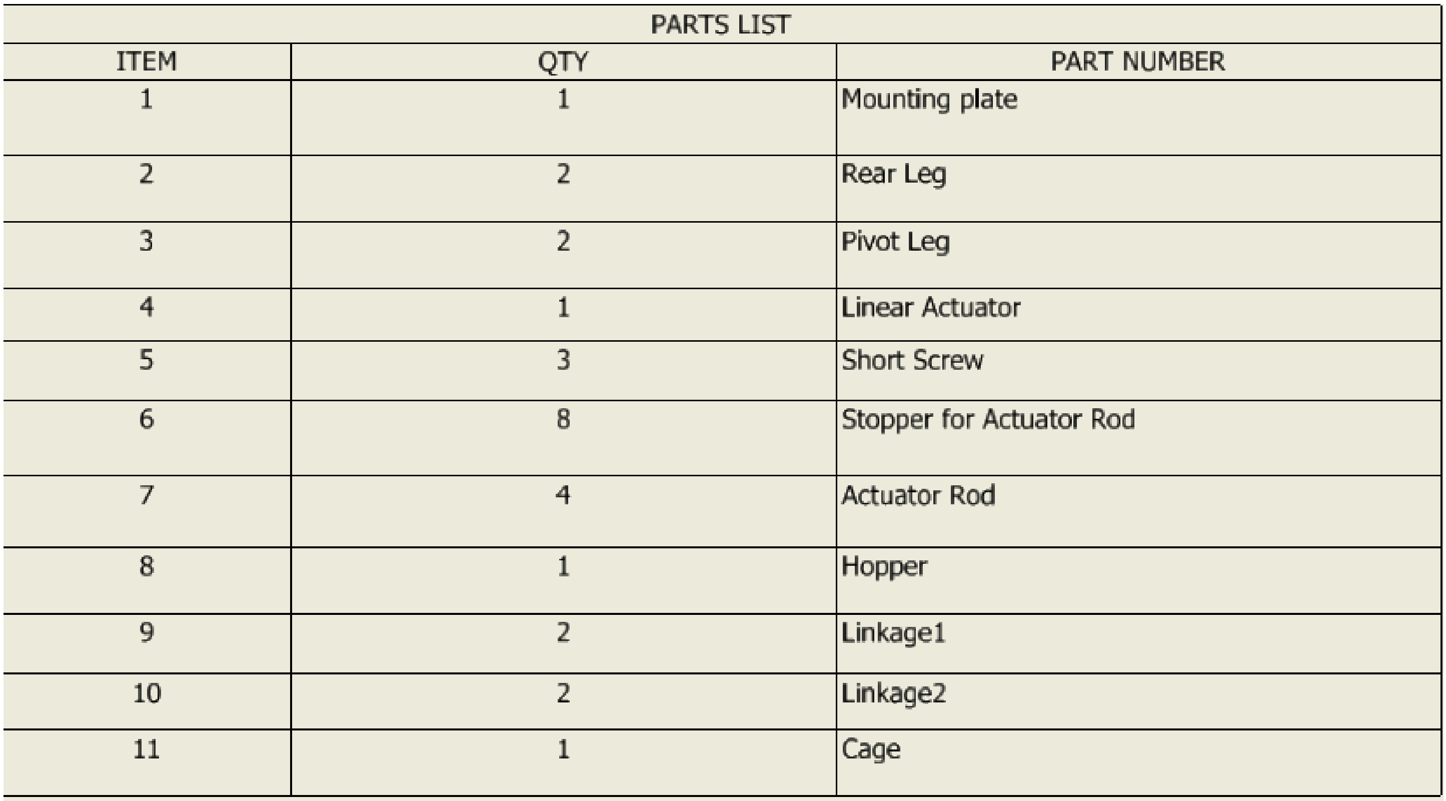
Exploded Assembly Drawing



PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	Mounting plate
2	2	Rear Leg
3	2	Pivot Leg
4	1	L near Actuator
5	3	Short Screw
6	8	Stopper for Actuator Rod
7	4	Actuator Rod
8	1	Hopper
9	2	L nkage1
10	2	L nkage2
11	1	Cage

DRAWN Fri-35	2021-03-06	
CHECKED		TITLE P3 Exploded Drawing
QA		
MFG		
APPROVED		SIZE D · DWG NO P3_Model_Final · REV
		SCALE 3/4 · SHEET 1 OF 1

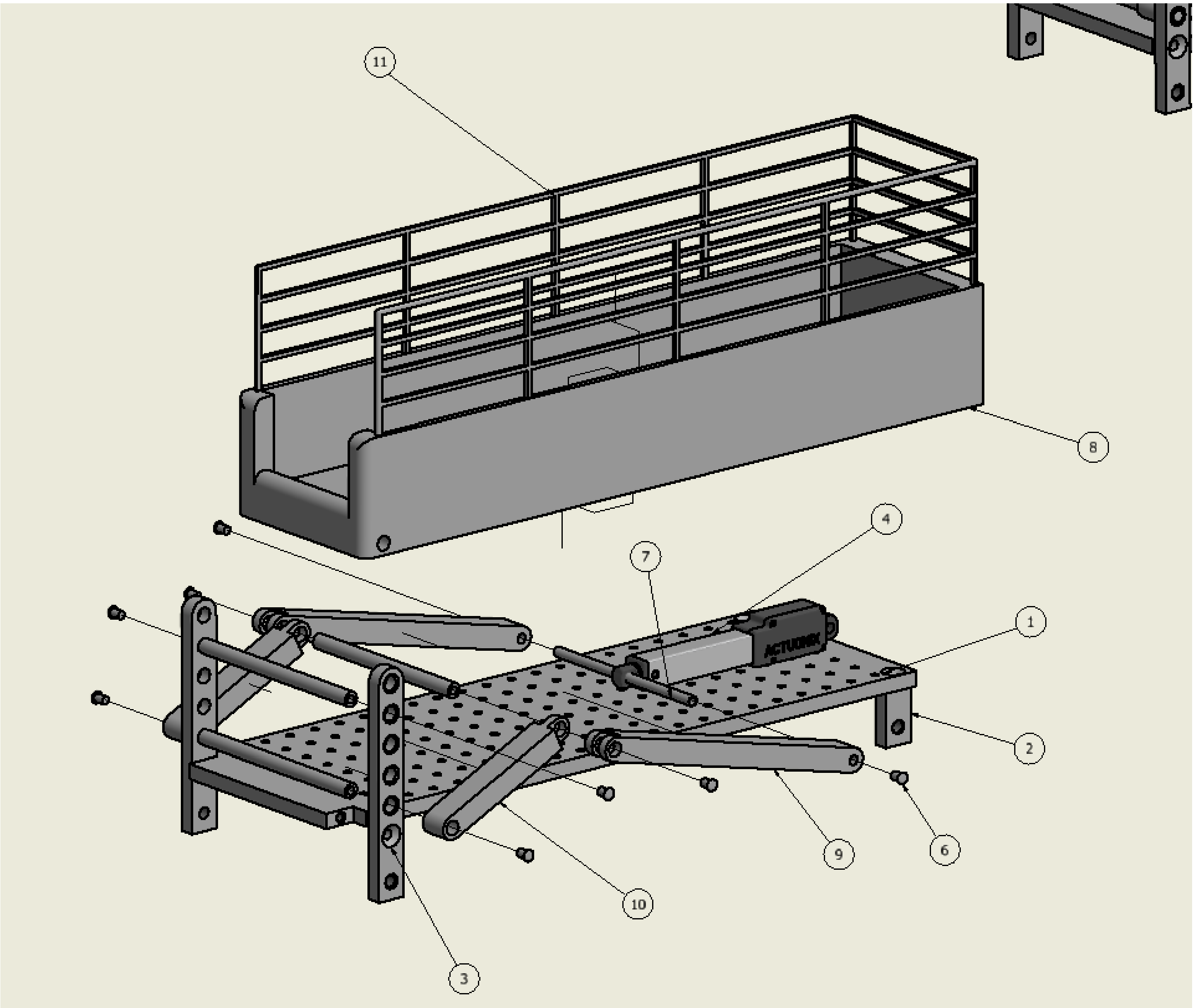
Parts List



PARTS LIST

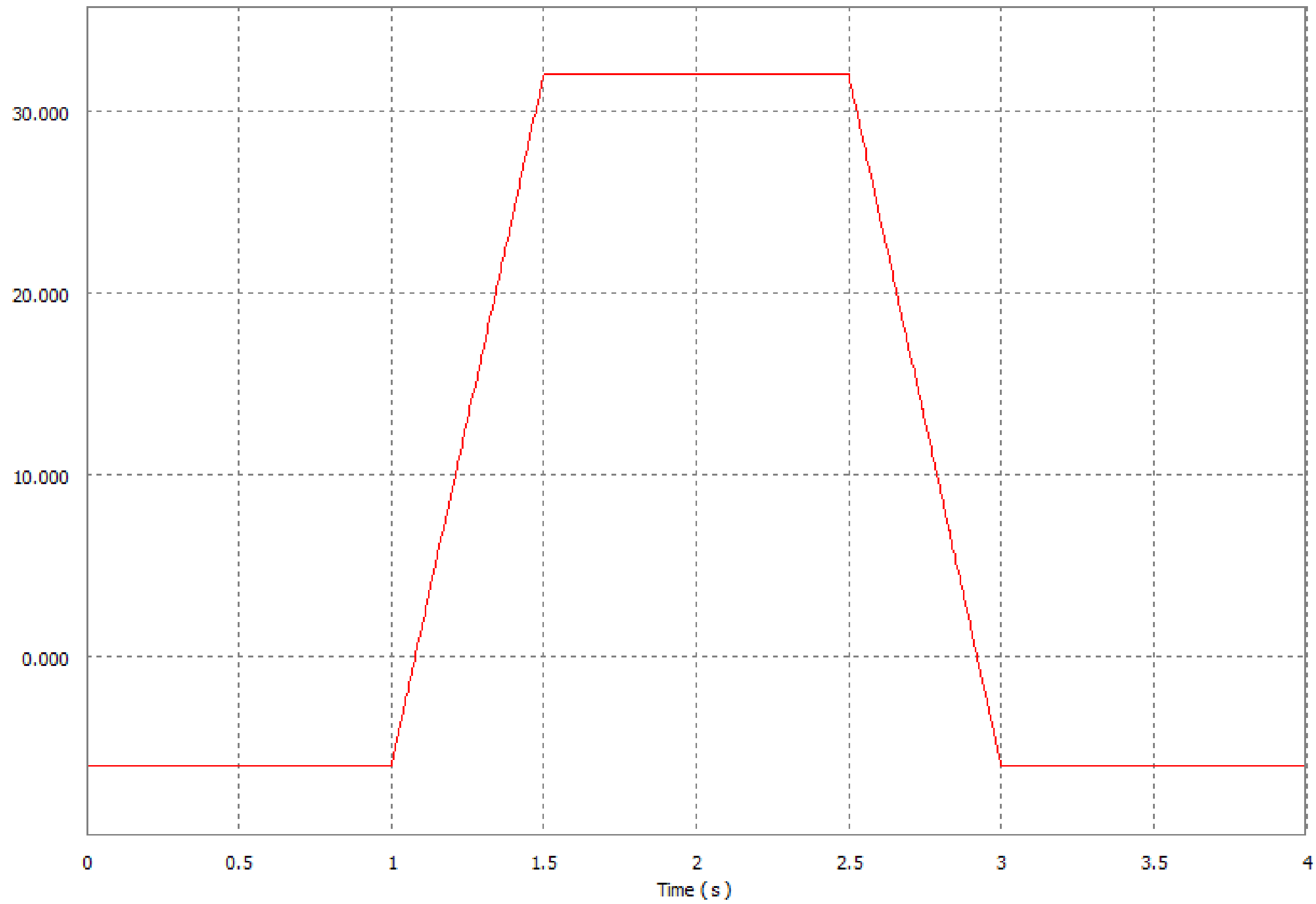
ITEM	QTY	PART NUMBER
1	1	Mounting plate
2	2	Rear Leg
3	2	Pivot Leg
4	1	Linear Actuator
5	3	Short Screw
6	8	Stopper for Actuator Rod
7	4	Actuator Rod
8	1	Hopper
9	2	Linkage1
10	2	Linkage2
11	1	Cage

Exploded Drawing Close-Up



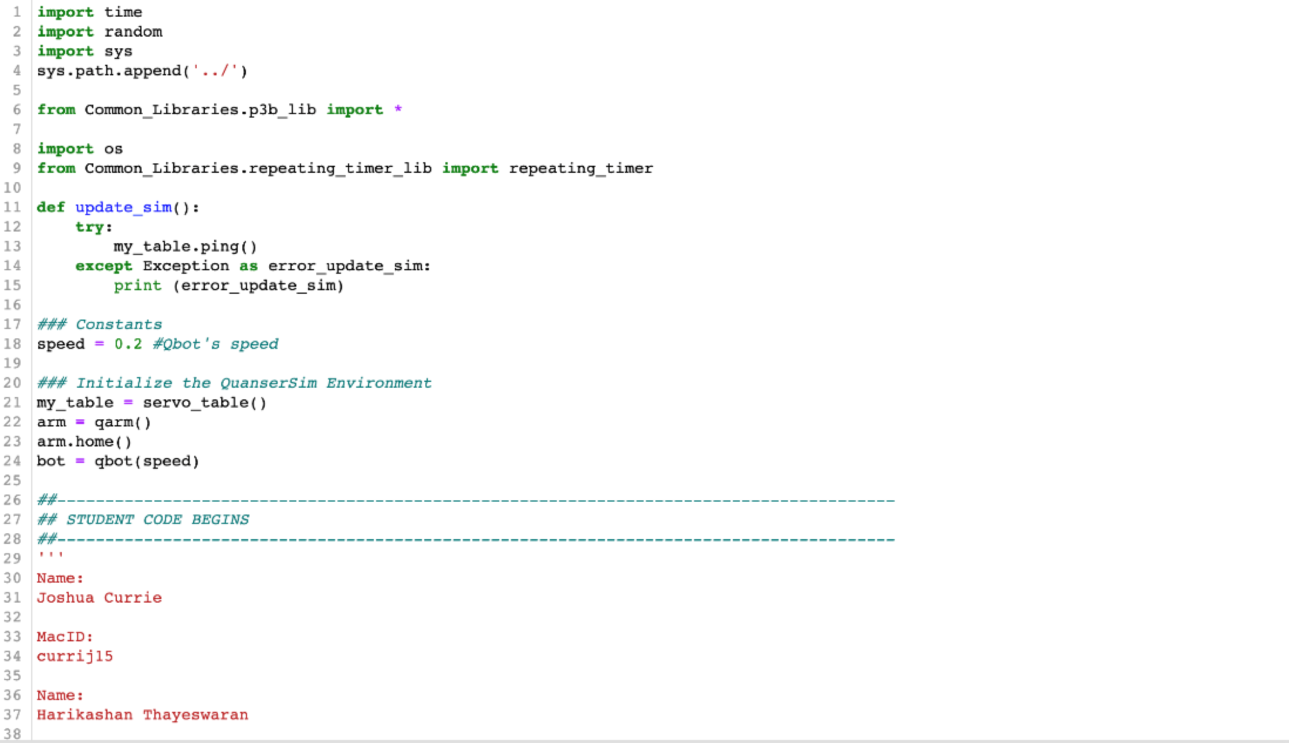
Appendix D – Output Grapher of Simulation

Output graph of angle of the hopper

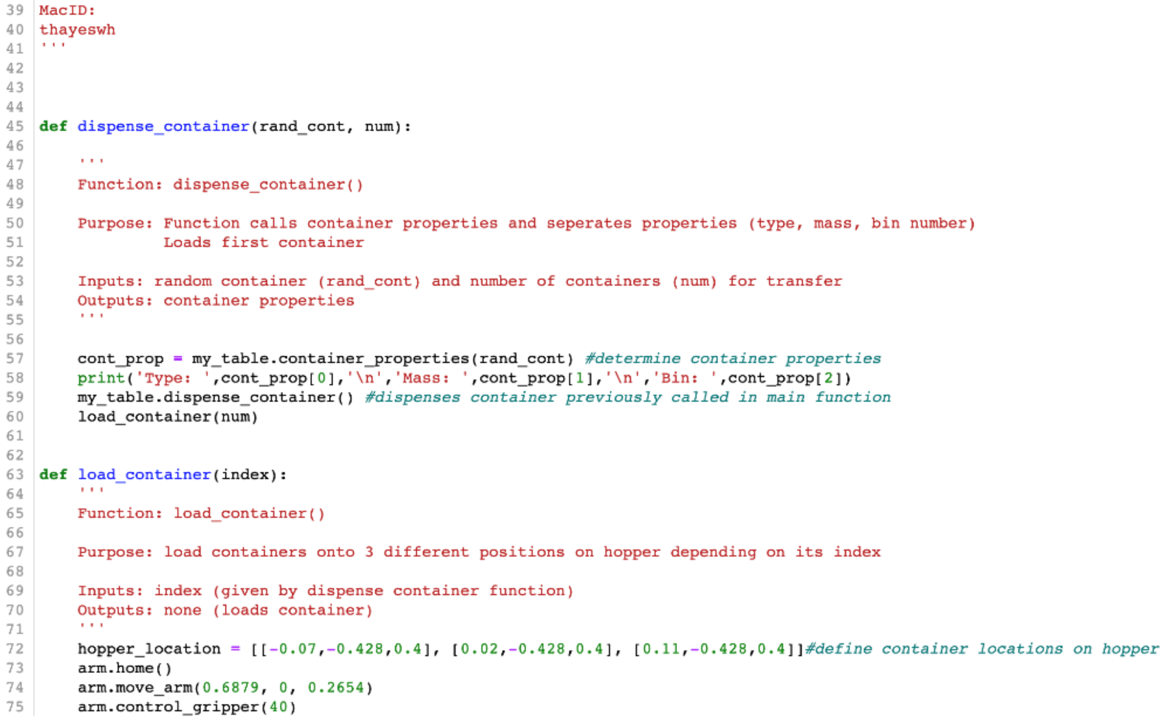


Appendix E – Screenshots of Computer Program:

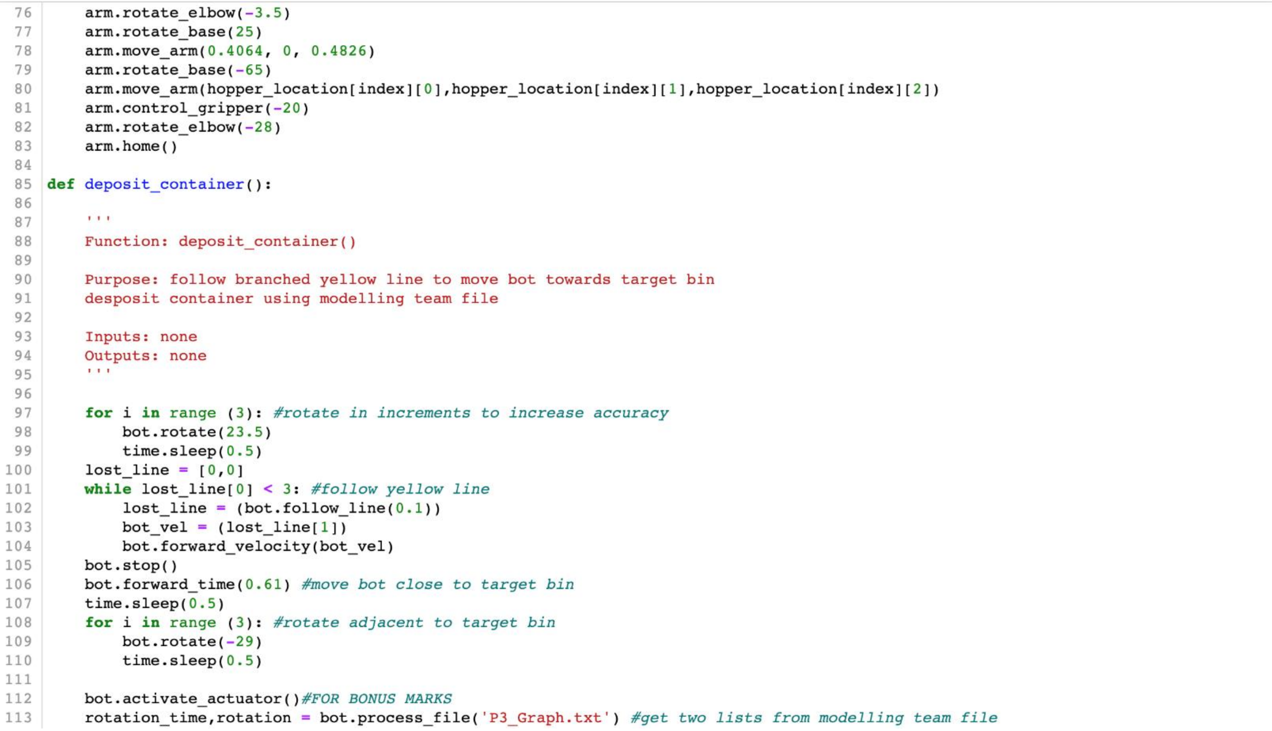
Screenshot of Computer Program



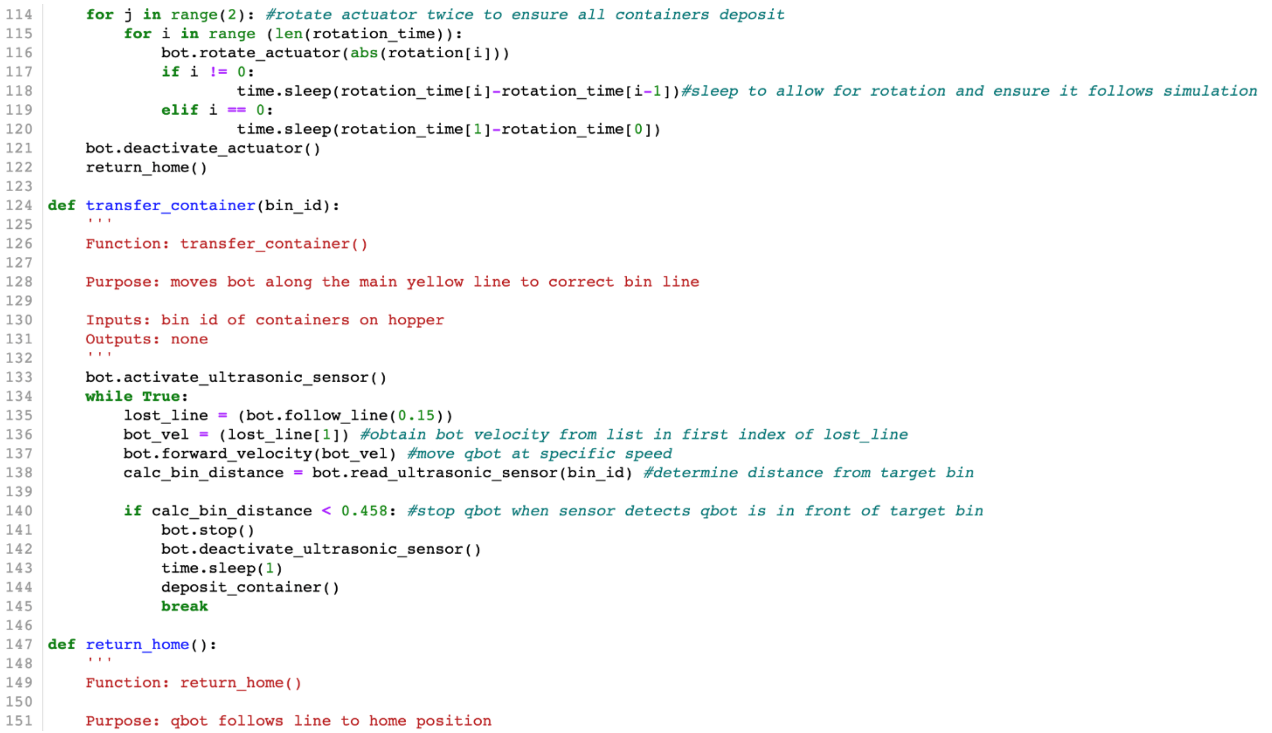
```python
import time
import random
import sys
sys.path.append('../')

from Common_Libraries.p3b_lib import *

import os
from Common_Libraries.repeating_timer_lib import repeating_timer

def update_sim():
    try:
        my_table.ping()
    except Exception as error_update_sim:
        print (error_update_sim)

### Constants
speed = 0.2 #Qbot's speed

### Initialize the QuanserSim Environment
my_table = servo_table()
arm = qarm()
arm.home()
bot = qbot(speed)

##---------------------------------------------------------------------------------------
## STUDENT CODE BEGINS
##---------------------------------------------------------------------------------------
'''
Name:
Joshua Currie

MacID:
currij15

Name:
Harikashan Thayeswaran

```



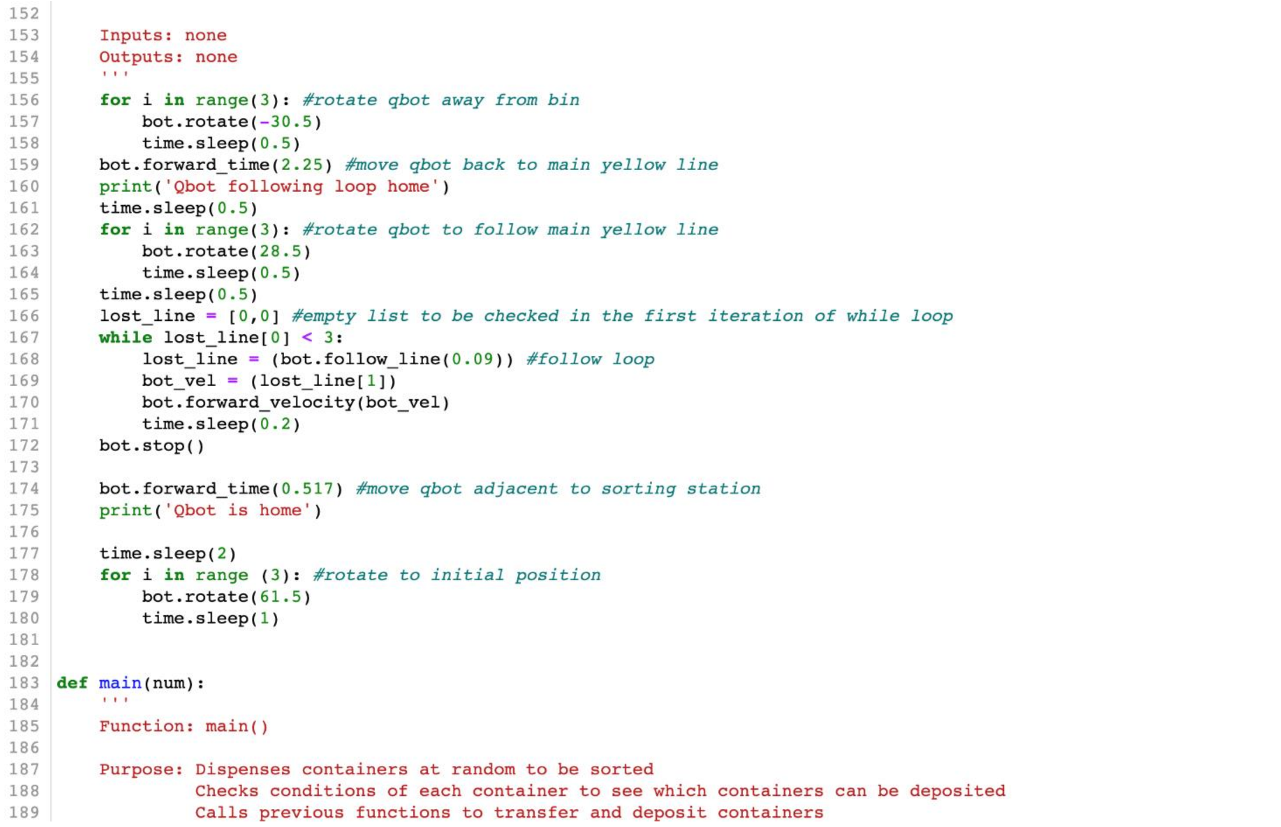
```python
MacID:
thayeswh
'''



def dispense_container(rand_cont, num):

    '''
    Function: dispense_container()

    Purpose: Function calls container properties and seperates properties (type, mass, bin number)
             Loads first container

    Inputs: random container (rand_cont) and number of containers (num) for transfer
    Outputs: container properties
    '''

    cont_prop = my_table.container_properties(rand_cont) #determine container properties
    print('Type: ',cont_prop[0],'\n','Mass: ',cont_prop[1],'\n','Bin: ',cont_prop[2])
    my_table.dispense_container() #dispenses container previously called in main function
    load_container(num)


def load_container(index):
    '''
    Function: load_container()

    Purpose: load containers onto 3 different positions on hopper depending on its index

    Inputs: index (given by dispense container function)
    Outputs: none (loads container)
    '''
    hopper_location = [[-0.07,-0.428,0.4], [0.02,-0.428,0.4], [0.11,-0.428,0.4]]#define container locations on hopper
    arm.home()
    arm.move_arm(0.6879, 0, 0.2654)
    arm.control_gripper(40)
```



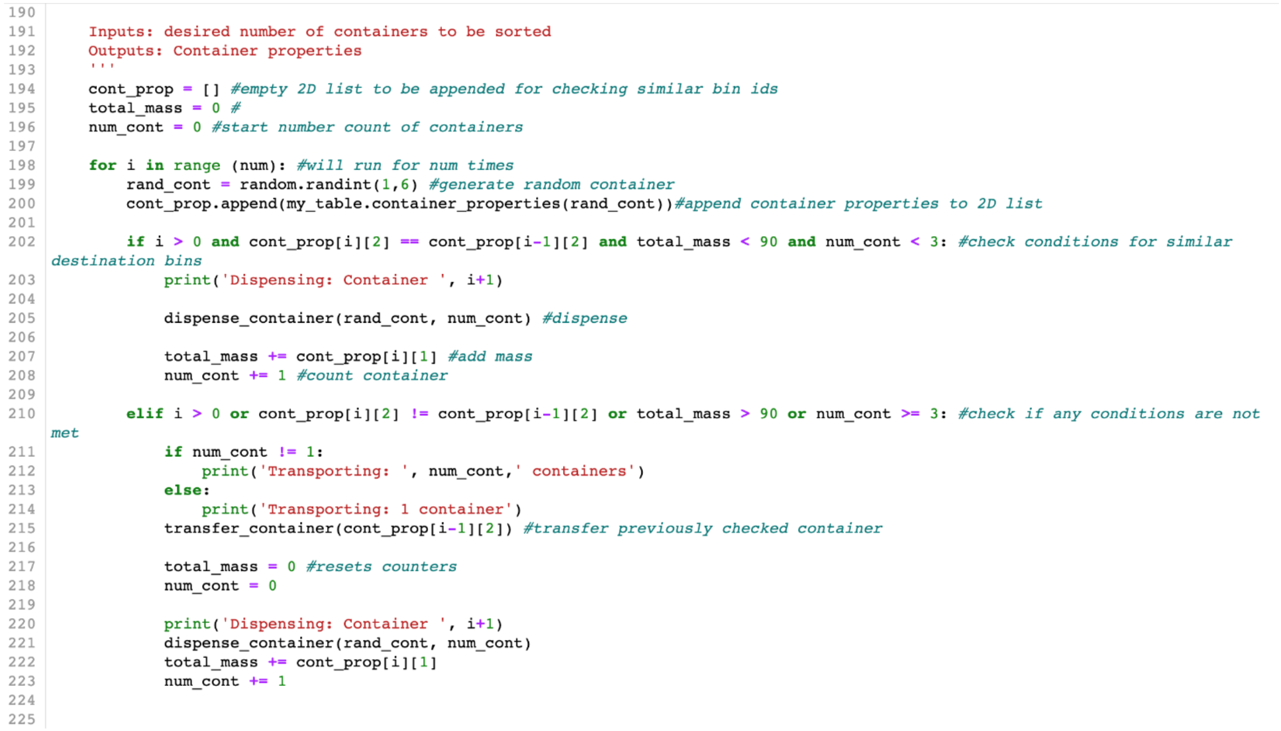
```
    arm.rotate_elbow(-3.5)
    arm.rotate_base(25)
    arm.move_arm(0.4064, 0, 0.4826)
    arm.rotate_base(-65)
    arm.move_arm(hopper_location[index][0],hopper_location[index][1],hopper_location[index][2])
    arm.control_gripper(-20)
    arm.rotate_elbow(-28)
    arm.home()

def deposit_container():

    '''
    Function: deposit_container()

    Purpose: follow branched yellow line to move bot towards target bin
    desposit container using modelling team file

    Inputs: none
    Outputs: none
    '''

    for i in range (3): #rotate in increments to increase accuracy
        bot.rotate(23.5)
        time.sleep(0.5)
    lost_line = [0,0]
    while lost_line[0] < 3: #follow yellow line
        lost_line = (bot.follow_line(0.1))
        bot_vel = (lost_line[1])
        bot.forward_velocity(bot_vel)
    bot.stop()
    bot.forward_time(0.61) #move bot close to target bin
    time.sleep(0.5)
    for i in range (3): #rotate adjacent to target bin
        bot.rotate(-29)
        time.sleep(0.5)

    bot.activate_actuator()#FOR BONUS MARKS
    rotation_time,rotation = bot.process_file('P3_Graph.txt') #get two lists from modelling team file
```



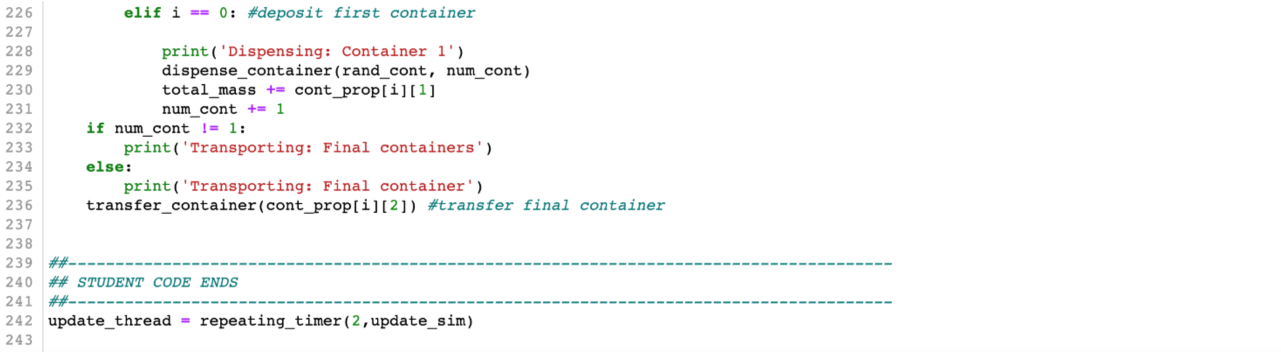
```
    for j in range(2): #rotate actuator twice to ensure all containers deposit
        for i in range (len(rotation_time)):
            bot.rotate_actuator(abs(rotation[i]))
            if i != 0:
                    time.sleep(rotation_time[i]-rotation_time[i-1])#sleep to allow for rotation and ensure it follows simulation
            elif i == 0:
                    time.sleep(rotation_time[1]-rotation_time[0])
    bot.deactivate_actuator()
    return_home()

def transfer_container(bin_id):
    '''
    Function: transfer_container()

    Purpose: moves bot along the main yellow line to correct bin line

    Inputs: bin id of containers on hopper
    Outputs: none
    '''
    bot.activate_ultrasonic_sensor()
    while True:
        lost_line = (bot.follow_line(0.15))
        bot_vel = (lost_line[1]) #obtain bot velocity from list in first index of lost_line
        bot.forward_velocity(bot_vel) #move qbot at specific speed
        calc_bin_distance = bot.read_ultrasonic_sensor(bin_id) #determine distance from target bin

        if calc_bin_distance < 0.458: #stop qbot when sensor detects qbot is in front of target bin
            bot.stop()
            bot.deactivate_ultrasonic_sensor()
            time.sleep(1)
            deposit_container()
            break

def return_home():
    '''
    Function: return_home()

    Purpose: qbot follows line to home position
```



```

    Inputs: none
    Outputs: none
    '''
    for i in range(3): #rotate qbot away from bin
        bot.rotate(-30.5)
        time.sleep(0.5)
    bot.forward_time(2.25) #move qbot back to main yellow line
    print('Qbot following loop home')
    time.sleep(0.5)
    for i in range(3): #rotate qbot to follow main yellow line
        bot.rotate(28.5)
        time.sleep(0.5)
    time.sleep(0.5)
    lost_line = [0,0] #empty list to be checked in the first iteration of while loop
    while lost_line[0] < 3:
        lost_line = (bot.follow_line(0.09)) #follow loop
        bot_vel = (lost_line[1])
        bot.forward_velocity(bot_vel)
        time.sleep(0.2)
    bot.stop()

    bot.forward_time(0.517) #move qbot adjacent to sorting station
    print('Qbot is home')

    time.sleep(2)
    for i in range (3): #rotate to initial position
        bot.rotate(61.5)
        time.sleep(1)


def main(num):
    '''
    Function: main()

    Purpose: Dispenses containers at random to be sorted
             Checks conditions of each container to see which containers can be deposited
             Calls previous functions to transfer and deposit containers
```

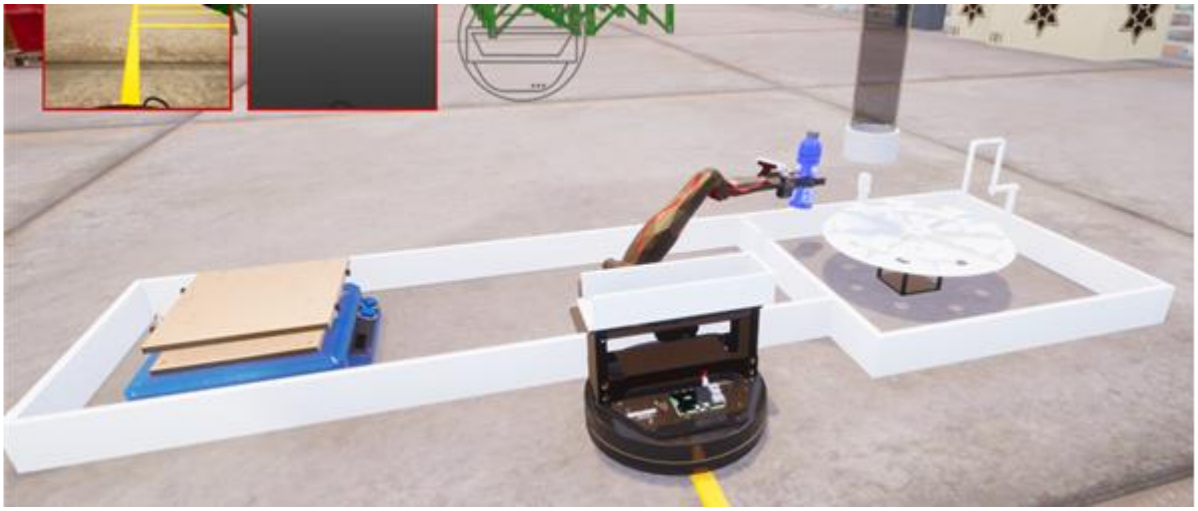


```

    Inputs: desired number of containers to be sorted
    Outputs: Container properties
    '''
    cont_prop = [] #empty 2D list to be appended for checking similar bin ids
    total_mass = 0 #
    num_cont = 0 #start number count of containers

    for i in range (num): #will run for num times
        rand_cont = random.randint(1,6) #generate random container
        cont_prop.append(my_table.container_properties(rand_cont))#append container properties to 2D list

        if i > 0 and cont_prop[i][2] == cont_prop[i-1][2] and total_mass < 90 and num_cont < 3: #check conditions for similar destination bins
            print('Dispensing: Container ', i+1)

            dispense_container(rand_cont, num_cont) #dispense

            total_mass += cont_prop[i][1] #add mass
            num_cont += 1 #count container

        elif i > 0 or cont_prop[i][2] != cont_prop[i-1][2] or total_mass > 90 or num_cont >= 3: #check if any conditions are not met
            if num_cont != 1:
                print('Transporting: ', num_cont,' containers')
            else:
                print('Transporting: 1 container')
            transfer_container(cont_prop[i-1][2]) #transfer previously checked container

            total_mass = 0 #resets counters
            num_cont = 0

            print('Dispensing: Container ', i+1)
            dispense_container(rand_cont, num_cont)
            total_mass += cont_prop[i][1]
            num_cont += 1

```



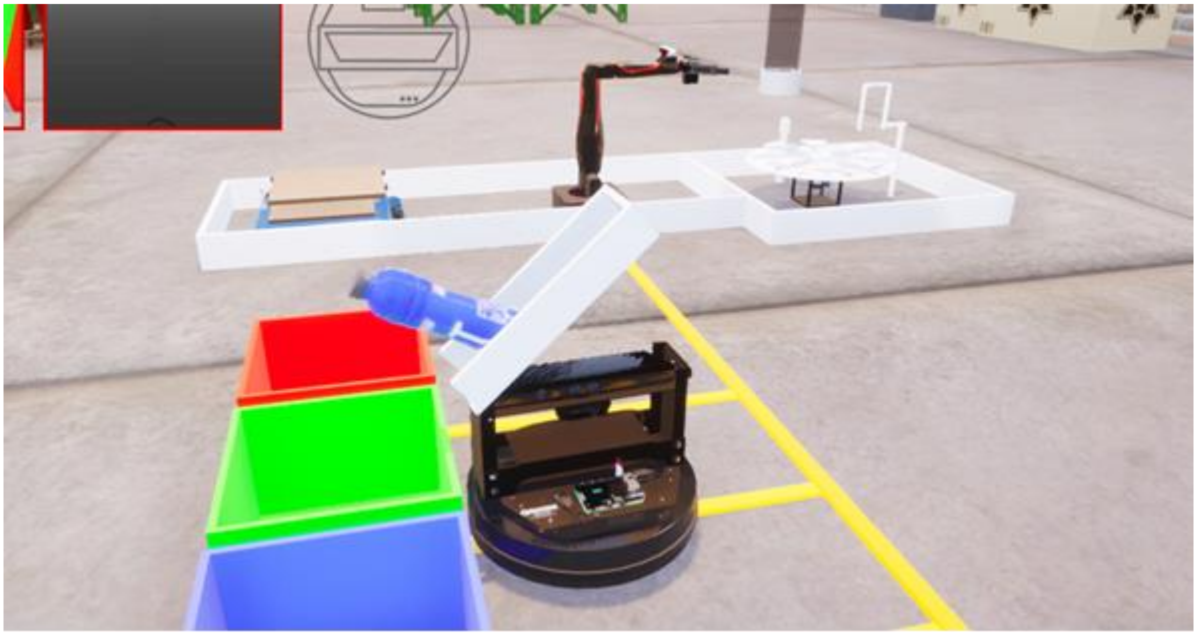
```
        elif i == 0: #deposit first container

            print('Dispensing: Container 1')
            dispense_container(rand_cont, num_cont)
            total_mass += cont_prop[i][1]
            num_cont += 1
    if num_cont != 1:
        print('Transporting: Final containers')
    else:
        print('Transporting: Final container')
    transfer_container(cont_prop[i][2]) #transfer final container


##---------------------------------------------------------------------------------------
## STUDENT CODE ENDS
##---------------------------------------------------------------------------------------
update_thread = repeating_timer(2,update_sim)

```

Loading Container



Q-Arm Placing Container on Hopper

Q-Bot Depositing Container



Q-Bot Following Loop to Return Home