Design Project 1 – Robotic End Effector System

ENGINEER 1P13 – Integrated Cornerstone Design Projects in Engineering

Tutorial: T06

Tues-52

Jennifer Desbarats (Desbaraj)

Kian Diggle (Digglek)

Shayan Siddiqui (Siddm47)

Habibah Aboueleinin (Aboueh2)

Shuja Wazir (Wazirs4)

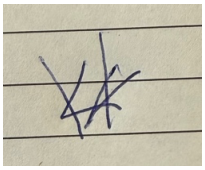
Submitted: December 3, 2025

Course Instructors: Dr. McDonald, Dr. Doyle, Dr. Ebrahimi, Dr. Fleisig, Dr. Hassan, Dr. Zurob

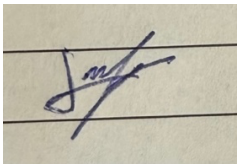
Table of Contents

Academic Integrity Statement

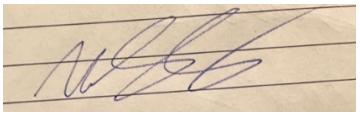
The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

 (Kian – 400617616 - Signature)

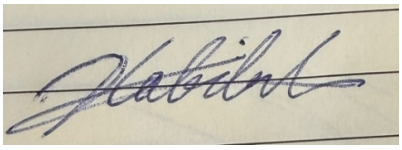
The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

 (Jennifer – 400617629 - Signature)

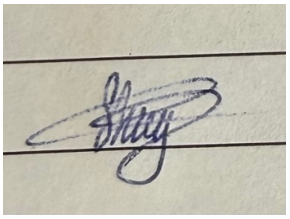
The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

 (Shayan – 4000617630 - Signature)

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.

 (Habibah – 400617643 - Signature)

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.

 (Shuja – 400617628 - Signature)

Executive Summary

The objective of this 1P13 Project was to facilitate the warehouse packaging process, where objects must be retrieved and packaged systematically. Warehouses often use robotic systems to transfer goods, but many fail to adapt to the variety of objects. This places a burden on warehouse facilitators, who must manually complete these tasks. This impedes the packaging process and can lead to a variety of injuries and physical strain. Additionally, the systems used in many warehouses pose an environmental impact. Typical mechanisms require a lot of material and energy to operate.

Our task was to create a robotic end-effector, to address issues posed by current warehouses. Objects were fabricated to simulate a warehouse environment, each with a varied shape. We used a programmable, robotic "Q-Arm" which had several joints and a servo motor. The setup included a packaging area where objects were transferred. Our team had to consider several factors to make the project a success. The mechanism had to accommodate a variety of shapes and sizes, grip securely onto each object, sort them effectively and execute a program specifically tailored for retrieving each.

Main Body

Strategies Used in Designing the Mechanism and Software

Our design was created through lots of prototyping, testing and refining. Each iteration was used for testing and improvement. This revealed both problems and solutions, leading us to new discoveries, such as the arm angle. The coding of the Q-Arm followed a similar process. We controlled the individual joints of the Q-Arm using specific commands and adjusted the angles through trial and error, checking where the gripper lands and refining the values until it reached each object consistently.

Iteration Process

Mechanism Iteration

Our mechanism went through different iterations throughout the design process. Our first design implemented a rack-and-pinion, this was because, "Rigid grippers, which rely on rigid mechanical components for the movement of their fingers (e.g., rack-and-pinion mechanisms). These are usually associated with higher gripping forces" [1]. The gears and rack were situated on a "base," which had a fixed arm sticking out horizontally. The rack had a matching arm, and the two met when the gears spun. This converted rotational motion of the gears into linear motion of the rack. The design also minimized friction, as the rack and gears sat atop the smooth base. However, when tested, we noticed that due to the Q-Arm position relative to the objects, the platform could not move towards them at a flat level.

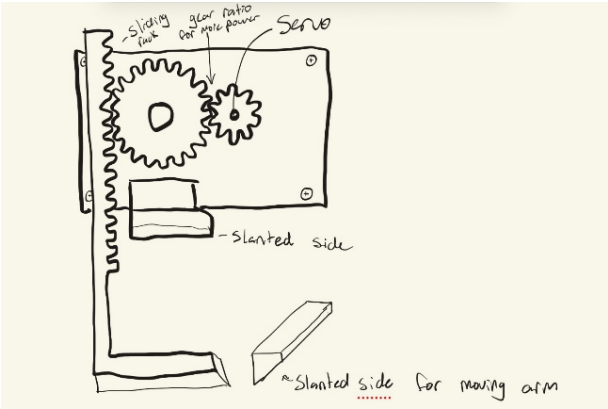


Figure 1. Finalized Mechanism Concept Sketch

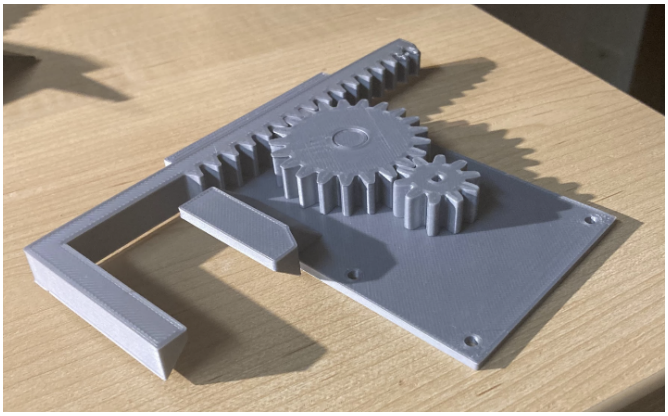


Figure 2. Initial Fabricated Design

This prompted another iteration, positioning the arms at a downward angle. Using a protractor, we measured the appropriate angle as the Q-arm approached the object, where the arms remained perpendicular to the floor, grasping objects in a straight motion. Additionally, we created a safety piece to prevent the actuator from spinning too far. When testing, we noticed that the 3D-printed mechanism was too smooth. According to our research, "materials, such as PLA, may be unsuitable due to their low coefficient of friction" [3] so without modifications it would be too slippery to pick objects up. We realized that we needed a sponge material that would form around the structure of the objects. This refinement led to higher consistency and ensuring less damage to objects.

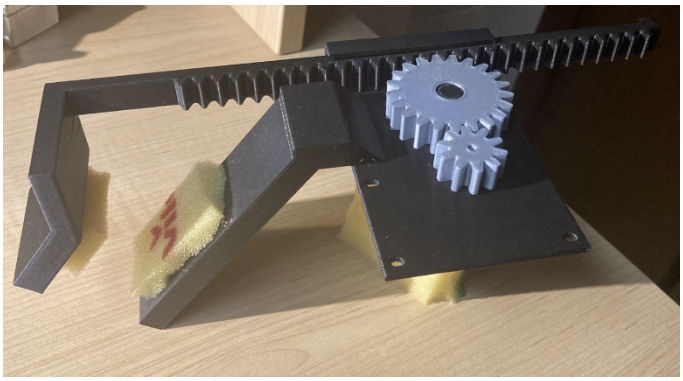


Figure 3. Mechanism with Angled Arms

Code Iteration

Overall, our individual code did not require much iteration or refinements except for adding colours and boldness to text output. For the Q-Arm code, we had to overcome several obstacles, including the servo motor spinning weakly. During the demonstration prep time, the Q-Arm assigned to us was different from the one we used during Design Studio, and we had to make some adjustments to our code as a result.

The Result of our Combined Solution

The end-effector and computing system combined to run the full warehouse process from logging in to packaging products. Users create an account or log in using sign_up() and authenticate(), then place orders with the barcode scanner. For each order, lookup_products() compares scanned items to the product list, ignores invalid entries, and sends the valid list to pack_products() to trigger the Q-Arm. complete_order() calculates the final total, prints a receipt, and saves the order, and customer_summary() produces a summary of total spending and items ordered when the user is finished.

During testing, our mechanism consistently picked up the sponge, bottle, rook, and D12 using angled arms and sponge pads (Figures 3–4). However, the witch hat and bowl were less consistent because of their tapered shapes.

Refinements for Future Iterations

With more time, we would focus on refinements that directly address what occured in testing. The witch hat and bowl were the least reliable objects, so we would reshape the ends of the arms to be slightly curved and adjust the height so that it clamps on the thicker part of the hat. For the bowl we would try to more consistently land on the bowl's ridge. We would also test alternate materials with more grip instead of just sponges.

From a computing standpoint, our main refinement would be changing how we control the arm. During testing, we only ever moved the individual joints of the Q-Arm. In future iterations we could rely on coordinate-based commands like set_arm_position() so that it could be implemented on different Q-Arms and be more consistent if objects moved.

Reflection of Group Work

Initially, we focused on group discussions emphasizing collaborating for every step but, eventually, we assigned specific lead roles for all group members based on individual strengths. One member oversaw integrating the functions and writing the main function, another designed the CAD and focused on the 3D-printing, someone else oversaw the Q-Arm code, and the other members took notes of the variations, detailed Team Milestones, and focused on putting together the report throughout the process. We consistently met to discuss progress during Design Studio, as well as additional meetings to ensure everyone had a strong understanding of all aspects of the project.

Reference List

[1] S. Cortinovis, G. Vitrani, M. Maggiali, and R. A Romeo, “Control methodologies for Robotic Grippers: A Review,” *Actuators*, vol. 12, no. 8, p. 332, Aug. 2023. doi:10.3390/act12080332

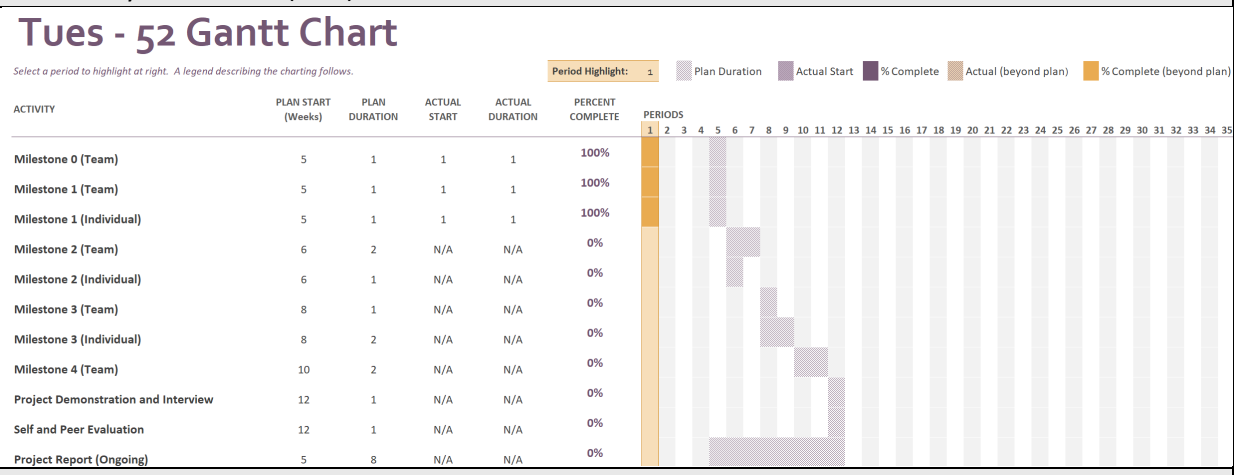
[2] M. Kundu, S. Pal, S. Bhuin, B. Hansda, and S. Das, “On the use of flexible materials for robot gripping application,” *Journal of the Association of Engineers, India*, vol. 84, no. 3-4, p. 79, Dec. 2014. doi:10.22485/jaei/2014/v84/i3-4/119871

[3] J. Suder, T. Kot, A. Panec, and M. Vocetka, “Analysis of increasing the friction force of the robot jaws by adding 3D printed flexible inserts,” *MM Science Journal*, vol. 2021, no. 6, pp. 522-5326, Dec. 2021. doi:10.17973/mmsj.2021_12_202112

Appendices

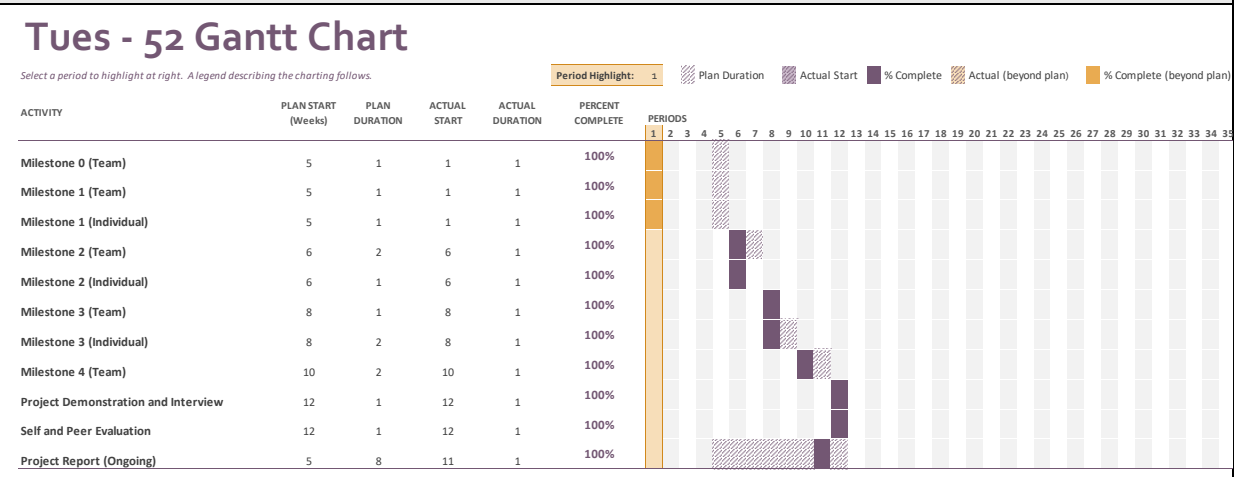
Appendix A: Project Schedule

Preliminary Gantt chart (Kian)



ACTIVITY	PLAN START (Weeks)	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0 (Team)	5	1	1	1	100%
Milestone 1 (Team)	5	1	1	1	100%
Milestone 1 (Individual)	5	1	1	1	100%
Milestone 2 (Team)	6	2	N/A	N/A	0%
Milestone 2 (Individual)	6	1	N/A	N/A	0%
Milestone 3 (Team)	8	1	N/A	N/A	0%
Milestone 3 (Individual)	8	2	N/A	N/A	0%
Milestone 4 (Team)	10	2	N/A	N/A	0%
Project Demonstration and Interview	12	1	N/A	N/A	0%
Self and Peer Evaluation	12	1	N/A	N/A	0%
Project Report (Ongoing)	5	8	N/A	N/A	0%

Final Gantt Chart (Shayan)



ACTIVITY	PLAN START (Weeks)	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0 (Team)	5	1	1	1	100%
Milestone 1 (Team)	5	1	1	1	100%
Milestone 1 (Individual)	5	1	1	1	100%
Milestone 2 (Team)	6	2	6	1	100%
Milestone 2 (Individual)	6	1	6	1	100%
Milestone 3 (Team)	8	1	8	1	100%
Milestone 3 (Individual)	8	2	8	1	100%
Milestone 4 (Team)	10	2	10	1	100%
Project Demonstration and Interview	12	1	12	1	100%
Self and Peer Evaluation	12	1	12	1	100%
Project Report (Ongoing)	5	8	11	1	100%

Logbook of Additional Meetings and Discussions (Jen)

Wednesday, October 22 at 6-7pm @ Thode

- Since moving forward with Jen's design, we concluded that there might be a lot of friction issues with her design. We also thought that the design was too complicated compared to other designs.
- As a team we changed to Kian's design which was much simpler and avoided those friction issues we found with Jen's design.
- As a team we discussed changes that could be made to make the system more effective.
- Adding slants on the grippers to help pick up the harder objects such as the witch hat and bowl.
- Redesign the CAD so we can begin printing next design studio.

Wednesday, November 12 at 6-7pm @ Thode

- Each member worked on individual code.
- Together we integrated our code into one while testing it out for any errors.
- Discussed how our current design has issues regarding the Q-arm constraints (How the arm can't be horizontally aligned with the objects) which for our current design makes it impossible to reach all objects.
- Suggested changing the arms of the grippers to be completely on an angle (Which we found using a protractor last design studio)
- Also removed the slanted arms after realizing it wasn't as effective.
- Redesigning the CAD so we can reprint at Hatch to be prepared for next design studio to test.

Wednesday, November 26 at 6-7pm @ Thode

- Discussing possible questions that could be asked for the interview next week
- Everyone presented and explained their code in detail and answered all questions

Why did we choose our design:

- Changed initial design because of too many moving parts and concerns about effective force
- Slanted arms were not effective to pick up objects (so we made them straight)
- Q arm constraints with how it comes in at an angle (we measured the desired angle and adjusted our pieces to match the angle and extended the arms)
- Small to big gear ratio for higher torque
- Arms were not grippy enough, so we added sponge on it to adapt the texture
- Arms initially were not long enough to open all the way for the sponge, so it had to be extended
- File off gears to avoid friction on servo
- We didn't want a mechanism that opened horizontally out of fear of it knocking over other objects
- With only one moving arm, more of the power is going into that arm which is needed for the smaller more precise pieces
- Had to add upper wall to the sliding arm so that it didn't fall over when it extended too far (center of mass was going to fall over the edge of the platform)

What we would do differently:

- Test out q arm to figure out how it moves before starting preliminary designs (knowing about the angle constraint before preliminary designs)
- Can remove the base piece and instead fasten everything to the platform

Splitting up document

- Sub sections
- Bullet points 1,2,4 are going to be split up into code and cad components for writing
- Bullet point 1: Shuja and Jen, 2: Kian, 3 – 4: Shayan, 5: Habibah

Monday December 1st – 4:30-5:30 @ PG

Review of CAD model

- Review how every individual piece was made
- Review different iterations of out CAD model

- Review all constraints used in the CAD model

What we would change if we had more time

- Grippier material
- Test out different iterations (one gear to rack – less friction but no small gear to big gear ratio)
- Make the base lighter (holes or just smaller platform)

Appendix B: Scheduled Weekly Meetings

Weekly Design Studio Agenda & Meeting Minutes	
Week 5 – Oct. 14	
Agenda: (Kian) • Complete Milestone 0 (TEAM) by the end of the day. • Each member complete Milestone 1 (INDIVIDUAL) by the end of the day. • Have TA Check-in and discuss based on feedback.	Notes and Action items: (Jen) • Adding initial problem based on prompt (sustainable, improving warehouse conditions) • focus on sustainability (adapt objectives and constraints to this focus) • Adapt constraints to keep warehouse worker in mind
Week 6 – Oct. 21	
Agenda: (Kian) • Review all members sketches and CADs. Discuss and choose the most effective design that we want to move forward with. • Have TA Check-in. • Each member complete Milestone 2 (INDIVIDUAL) by the end of the day. • Work on Milestone 2 (TEAM) to be completed before the next design studio.	Notes and Action items: (Jen) • Moving forward with Jen's design. • Will change many parts in the design to increase effectivity such as the base piece to , rack, and sliding piece. • Implement feedback given by the TA such as adding a gap to reduce friction.
Week 7 – Oct. 28	
Agenda: (Kian) • Begin printing our design. • Have TA Check-in. • Discuss as a group if changes need to be made regarding Q-arm constraints.	Notes and Action items: (Jen) • Transition to Kian's design • Test out the claw. • Get started on Q-Arm code. • Think of possible improvements to the design.
Week 8 – Nov. 4	
Agenda: (Kian) • Each member work on Milestone 3 (INDIVIDUAL) to be completed before the next design studio. • Work on Milestone 3 (TEAM) to be completed before the next design studio. • Have TA Check-in. • Test claw and starting Q-arm code.	Notes and Action items: (Jen) • Need to implement feedback given by TA regarding both pack_products() and main() functions. • Make changes on design to accommodate for the Q-arm constraints.

Week 9 – Nov. 11	
Agenda: (Kian) • Each member work on Individual code. • Have TA Check-in. • Complete Computing Work Period (TEAM) by the end of the day.	Notes and Action items: (Jen) • Finish Individual code functions • Integrate individual functions together. • Continue working on team functions
Week 10 – Nov. 18	
Agenda: (Kian) • Work on Milestone 4 (TEAM) to be completed before the next design studio. • Test new claw and work on Q-arm code. • Have TA Check-in.	Notes and Action items: (Jen) • Reprint fixed mechanism with longer arm. • Complete Q-arm code for each object. • Fix gears to prevent getting stuck.
Week 11 – Nov. 25	
Agenda: (Kian) • Test out new and improved claw. • Continue perfecting Q-arm code for each object. • Have TA Check-in.	Notes and Action items: (Jen) • All members review code and ensure we are thoroughly prepared for the upcoming interview.

Appendix C: Comprehensive List of Sources

[1] S. Cortinovis, G. Vitrani, M. Maggiali, and R. A Romeo, “Control methodologies for Robotic Grippers: A Review,” *Actuators*, vol. 12, no. 8, p. 332, Aug. 2023. doi:10.3390/act12080332

[2] M. Kundu, S. Pal, S. Bhuin, B. Hansda, and S. Das, “On the use of flexible materials for robot gripping application,” *Journal of the Association of Engineers, India*, vol. 84, no. 3-4, p. 79, Dec. 2014. doi:10.22485/jaei/2014/v84/i3-4/119871

[3] J. Suder, T. Kot, A. Panec, and M. Vocetka, “Analysis of increasing the friction force of the robot jaws by adding 3D printed flexible inserts,” *MM Science Journal*, vol. 2021, no. 6, pp. 5322-5326, Dec. 2021. doi:10.17973/mmsj.2021_12_202112

[4] Singh, J., Srinivasa, k., and Singh, J., "Selection of Gear Ratio for Smooth Gear Shifting," *SAE Technical Paper* 2012-01-2005, 2012, doi:https://doi.org/10.4271/2012-01-2005.

[5] Vaishya , M., and Singh, R. (June 11, 2003). "Strategies for Modeling Friction in Gear Dynamics ." ASME. *J. Mech. Des*. June 2003; 125(2): 383–393. doi: https://doi.org/10.1115/1.1564063

Table 1 Material Properties

Material	**Density (g/cm^3)**	**Yield Strength (MPa)**
3D-Printed PLA Plastic	0.8	60
Sponge	15	50
Steel M3 Hex Nut	7.85	640
Steel M3x16 Pan Head Screw	7.85	640

Appendix D: Additional Documentation

CAD Part Model Images:

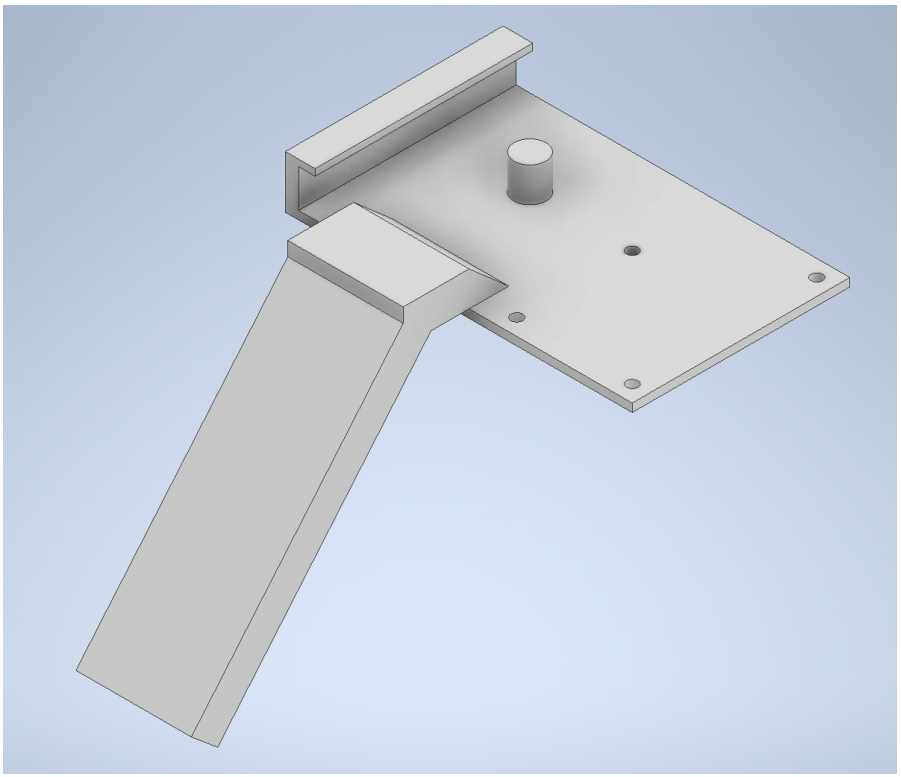


Figure 5. Base

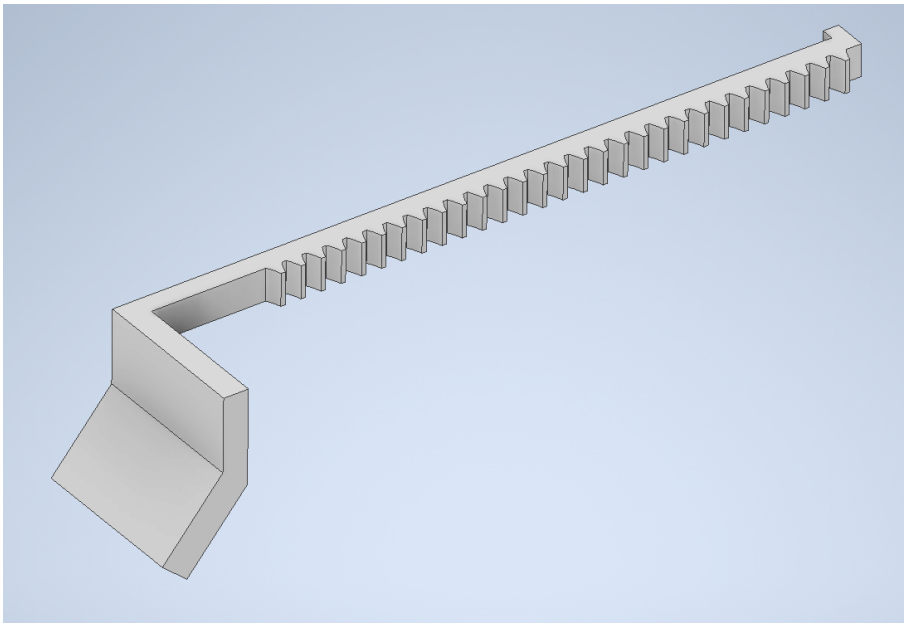


Figure 6. Rack

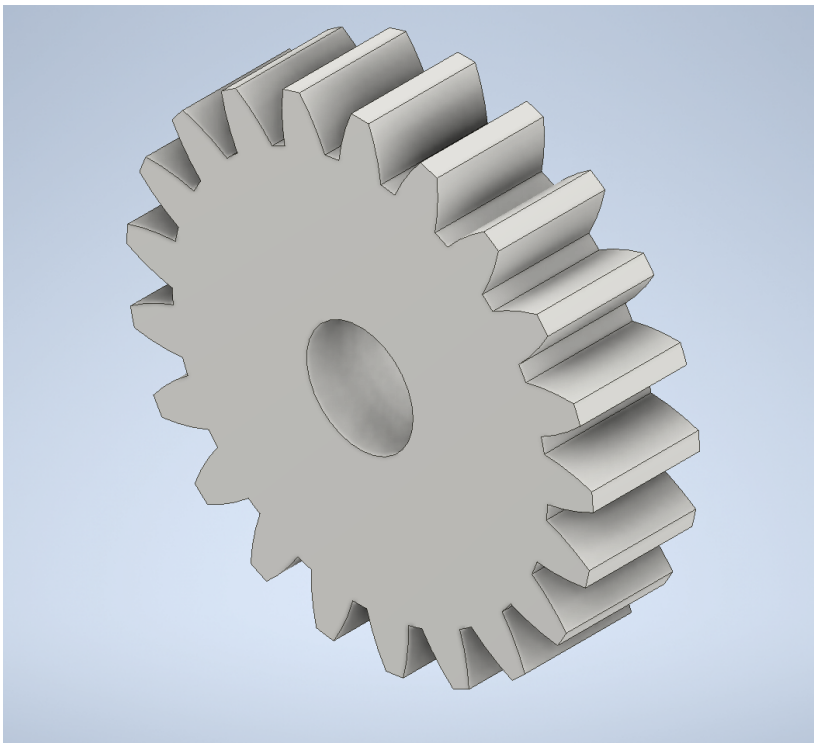


Figure 7. Large Spur Gear

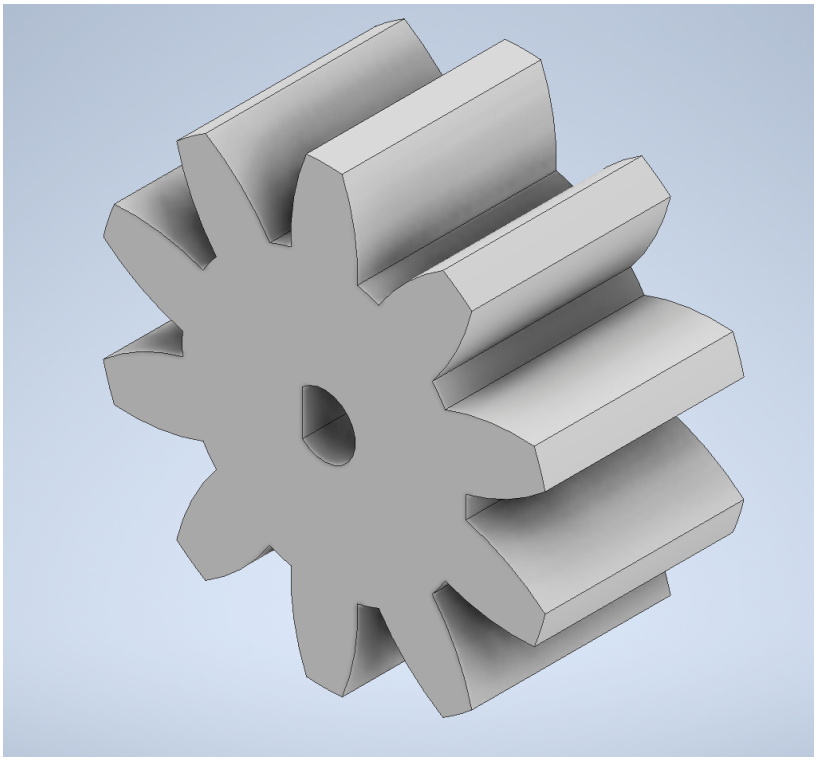


Figure 8. Small Spur Gear

Overall Mechanism Assembly:

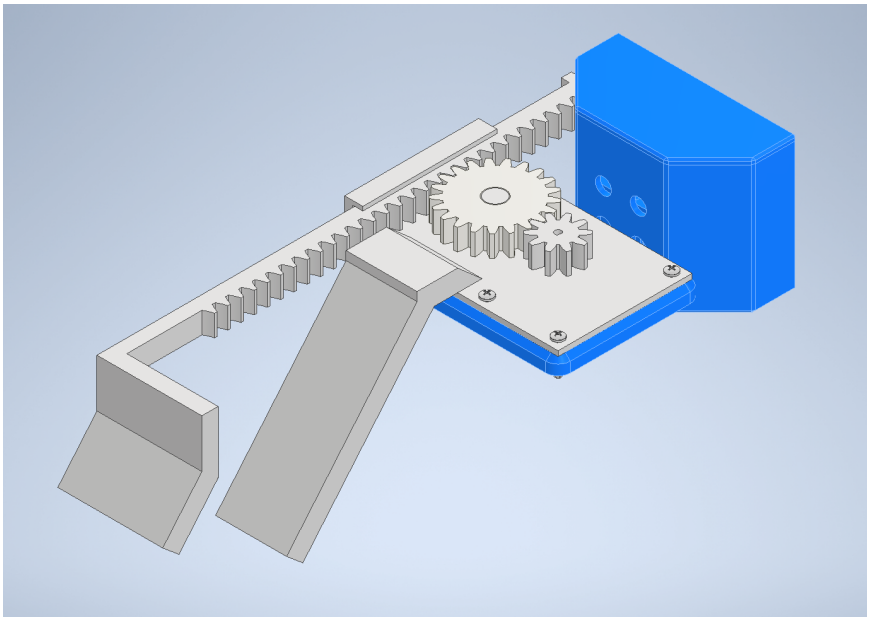


Figure 9. Mechanism Assembly Model

Mechanism Exploded View Drawing:

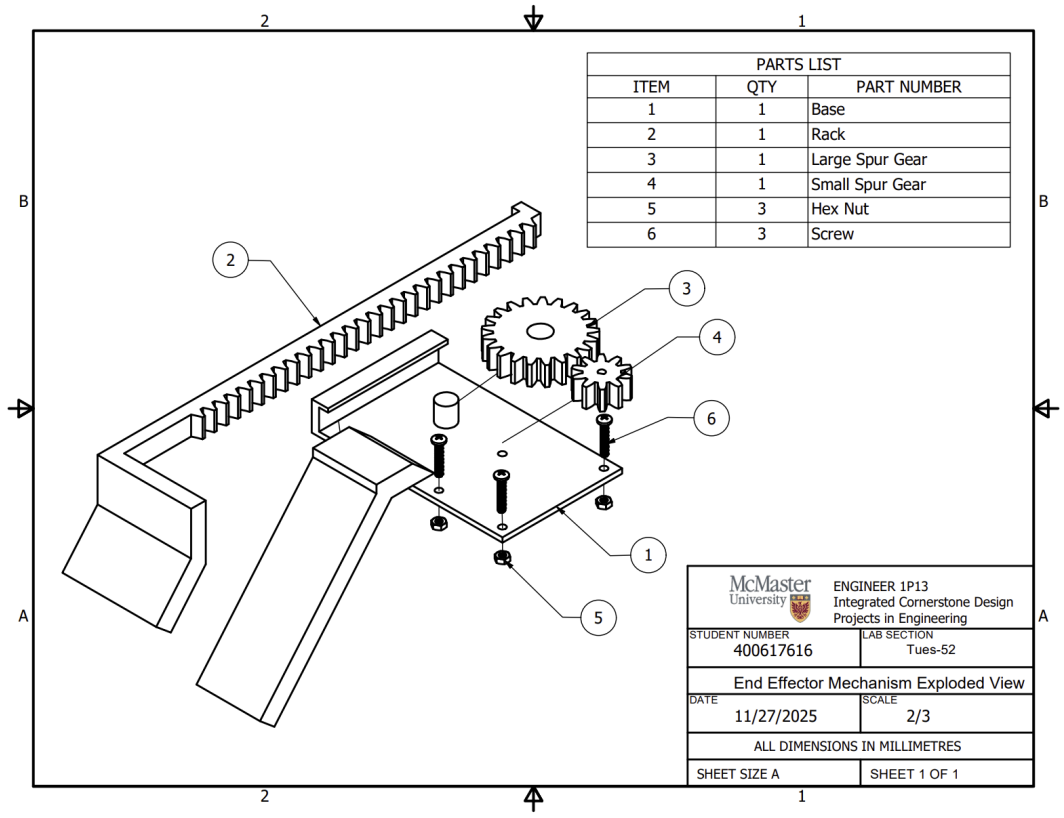


Figure 10. Mechanism Exploded View

Dimensioned Engineering Drawings:

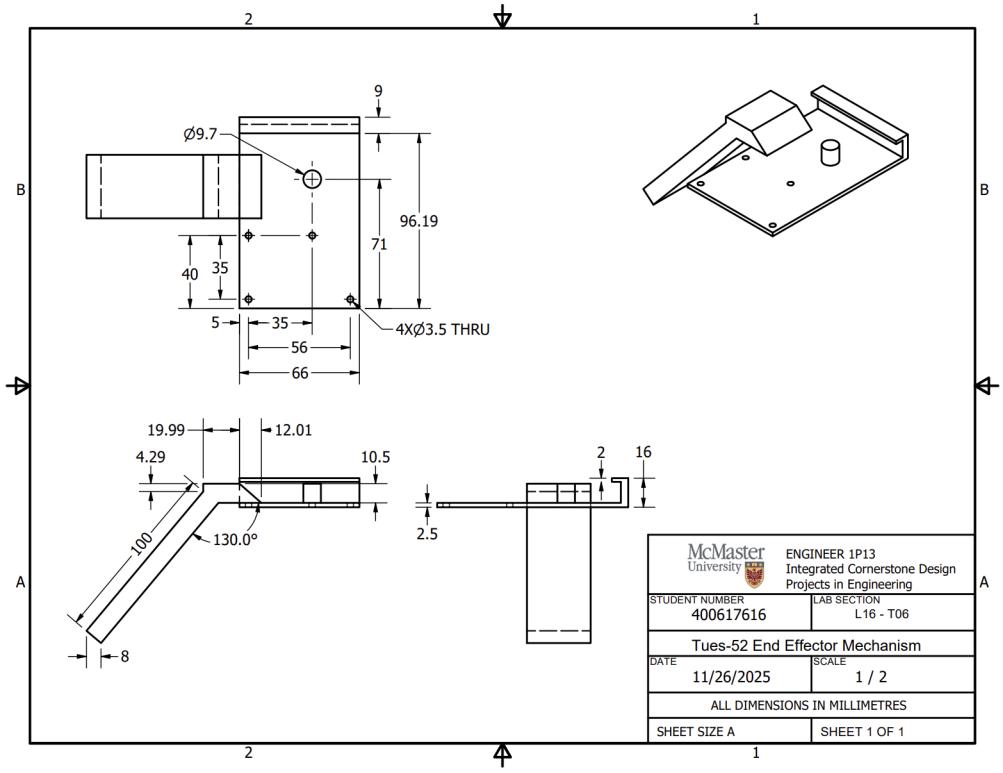


Figure 11. Base Drawing

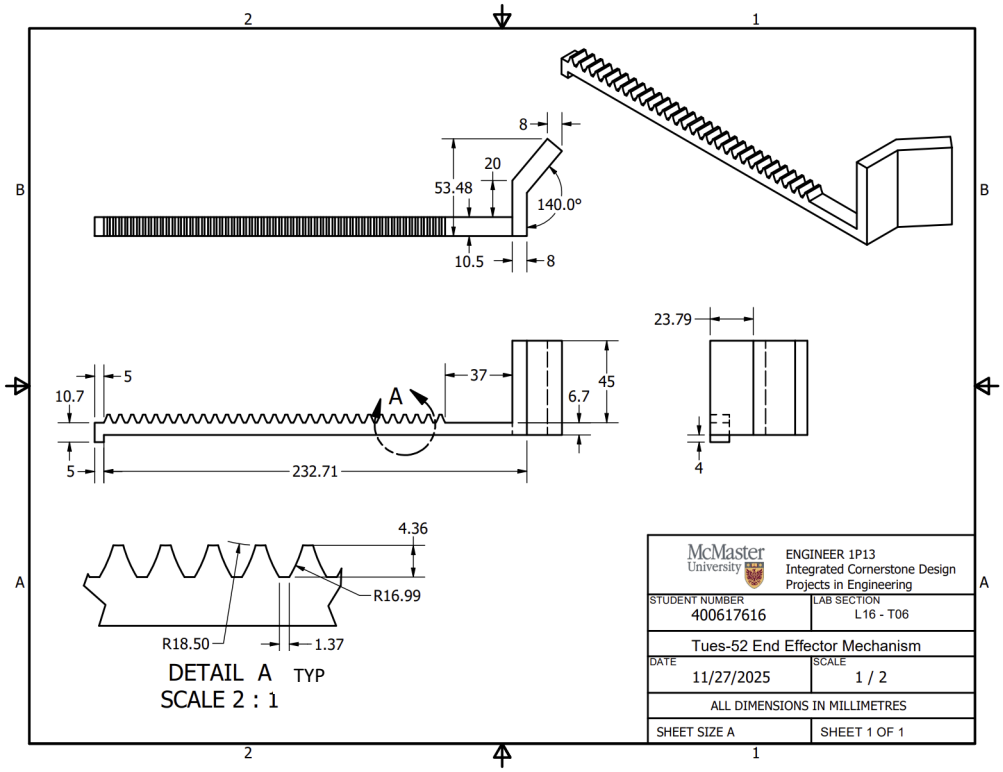


Figure 12. Rack Drawing

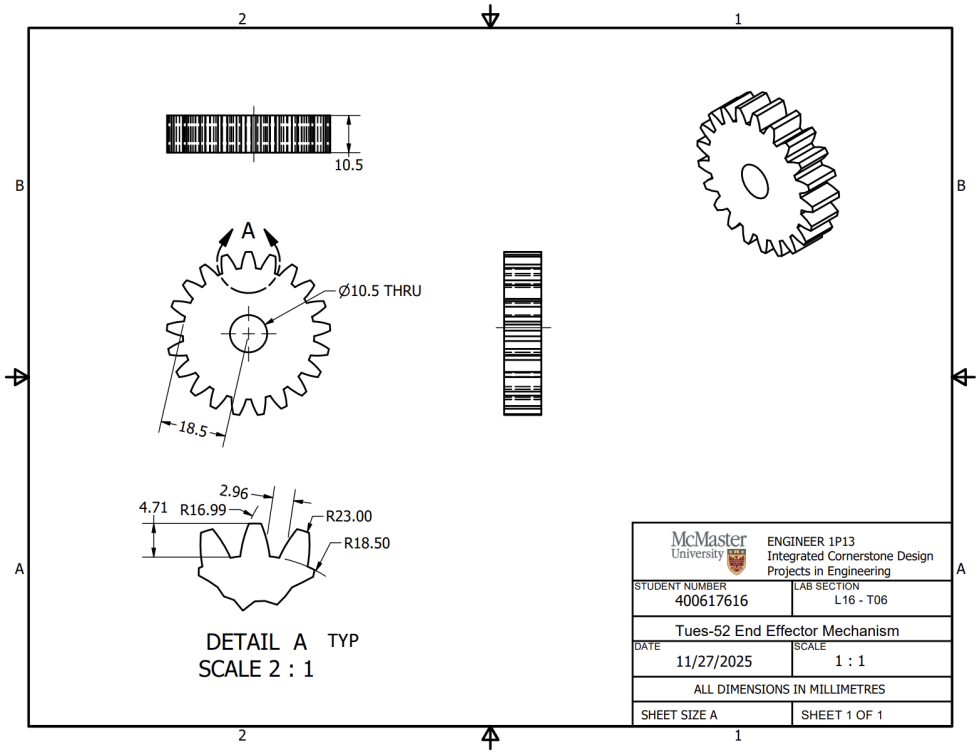


Figure 13. Large Spur Gear Drawing

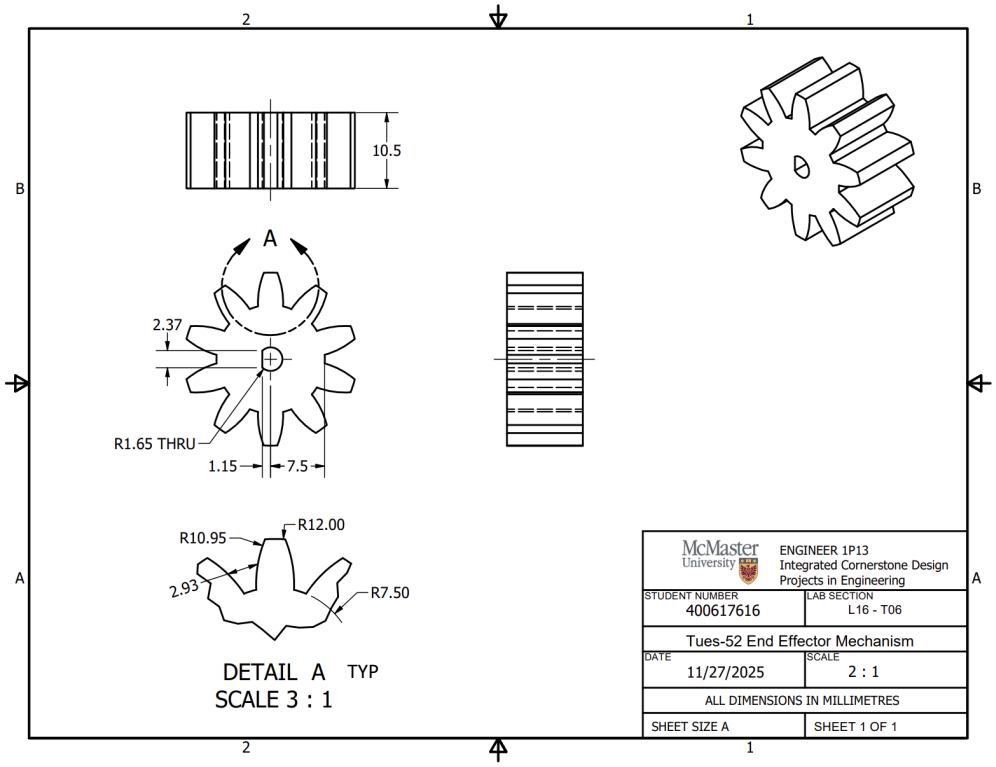


Figure 14. Small Spur Gear Drawing

Completed Code:

```
"""
This program simulates a warehouse program that allows the user to sign in or create an account, then process orders of various object and print a summary of all orders.

Athors: Jennifer Desbarats, Kian Diggle, Habibah Abouleinin, Shayan Siddiqui, Shuja Wazir
Date: December 3rd 2025
"""


import csv
import bcrypt
import random

##sign up function
def sign_up():
    """
    Shuja Wazir
    The sign up function is in charge of creating new accounts. It needs to check if the account already exists and if it does then it calls the authenticate function. If it doesnt exist it gets the user to create a username and adds it to  a csv file. It also ensures the user has a strong password meeting all the password specifications.
    """
    #Opening File
    file = open("users.csv")

    #Intialising Variables for userid
    no_repeat = False #we dont want repeating characters
    names = [] #List of usernames

    #Asking for Input
    userid = input("Enter New UserID: ")
    while no_repeat == False:

        for line in file:
            word = line.split(',') #split each line into the username and the password
            names.append(word[0]) #creates a list of all the names that already exist in the file

    #Checking if userid already exists
        if not (userid in names): #if the user ID does not already exist in the list of usernames
            no_repeat = True #we can say that the userid doesnt exist
        else:
            print("\033[1;31mERROR : This userid already exists\033[0m") # ask the user if they already have an account and would like to sign in
            check_account = input(("if you already have an account, enter y to sign in, enter anything else to continue: ")).lower()
            if check_account == "y":
                authenticate() #run the authenticate function (sign in)
                return
            else: #if they dont already have an account
                userid = input("Enter New UserID: ") #they need a unique user ID so they must enter a new one

    file.close()
```

Figure 15. Beginning of sign up function

```
    #If it doesn't exist then ask for a password
    if no_repeat == True:
        while True:
            #Initialising all the variables (all the chriteria that the password must have -> they all need to be true for the password to be accepted)
            length = False
            upper = False
            lower = False
            digit = False
            symbol = False
            space = False
            symbols = ["!","-",",","@","#","$","%","^","&","*","(",")","_","[","]"]
            password = input("Enter Your Password (Ensure it has an uppercase, lowercase, one digit, one symbol and at least 6 characters): ")
            characters = list(password) #make a list with each password character

        #Checking for conditions of the password
            for char in characters: #for each character in the password
                if char.isupper():
                    upper = True
                if char.islower():
                    lower = True
                if char.isdigit():
                    digit = True
                if len(password) > 5:
                    length = True
                if char.isspace():
                    space = True
                if char in symbols:
                    symbol = True

        #Printing out the error statements if the password doesnt meet specifications
            if not upper:
                print("\033[1;31m\nERROR : Please include an uppercase letter\033[0m\n")
            if not lower:
                print("\033[1;31m\nERROR : Please include a lowercase letter\033[0m\n")
            if not digit:
                print("\033[1;31m\nERROR : Please include a number\033[0m\n")
            if not length:
                print("\033[1;31m\nERROR : Please extend the length of your password to 6 characters\033[0m\n")
            if space:
                print("\033[1;31m\nERROR : Please do not include spaces in your password\033[0m\n")
            if not symbol:
                print("\033[1;31m\nERROR : Please include one of these symbols, (  !.@#$%^&*()_[]  )\033[0m\n")

            if upper and lower and digit and length and symbol and not space:
                break

        #Writing user and password down in the csv file

        hash =  bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt() ).decode('utf8')

        file = open("users.csv", mode='a')
        file.write("\n" + userid + "," + hash)
        file.close()
```

Figure 16. End of sign up function

```python
##authenticate function

def authenticate():
    """
    Jennifer Desbarats

    The authenticate function is the first function ran. It asks the user if they already have an account, if they do then it asks for their username and password to log them in. Or else it calls the sign up function to create an account
    """

    #ask if they already have an account
    existant_account = input("Do you already have an account? (y/n): ").lower()


    # if they typed anything other than y or n
    while existant_account != "y" and existant_account != "n": #infinite loop until input is correct
        print("\033[1;31mInvalid Input\033[0m\n")
        existant_account =  input("Do you already have an account? (y/n): ").lower()

    # if they type "n" they need to sign up first
    if existant_account == "n":
        sign_up()  #call the sign_up() function to create an account
        print("Please sign in using your new user ID\n")


    # ask for username and password
    logged_in = False  # start as not logged in

    while logged_in == False:  # keep looping until they are log in
        userid = input("Enter your username: ")
        password = input("Enter your password: ")

        # open users.csv to read saved usernames and hashed passwords
        user_found = False
        with open("users.csv", "r") as file:

            for row in file:
                words = row.split(",")
                if len(words) == 2: #skip empty lines or lines with too many entries
                    stored_userid = words[0] #words 0 is the username
                    stored_hash = words[1].strip() #stored hash is the stored password (words[1])

                    # check if username matches
                    if userid == stored_userid:
                        user_found = True
                        # check if password matches
                        if bcrypt.checkpw(password.encode('utf-8'), stored_hash.encode('utf-8')):
                            print("\033[1;32mLogin successful!\033[0m")
                            logged_in = True
                            break #exit the while loop
                        else:
                            print("\033[1;31mIncorrect password. Try again.\033[0m\n") #keep looping if the username is found (user_found = True) but the password is incorrect

            if user_found == False: #the username didnt exist
                print(" Username not found. Try again.\n") #it will keep looping through the while loop as logged_in is still false

    return userid
```

Figure 17. Authenticate function

```python
##Customer summary
def customer_summary(userid):

    """
    Habibah Abouleinin

    Customer summary summarises all orders completed by the customer.  It prints a proffesional reciept and keeps track of the total number of orders completed by the customer.
    """
    filename = 'orders.csv'
    total_orders = 0
    total_spent = 0
    product_counts = {}

    # Read each order (one per line)
    with open(filename, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            if len(row) >= 2 and row[0] == userid: #if the userid matches the current users user id
                total_orders += 1
                total_spent += float(row[1])
                for product in row[2:]:
                    if product not in product_counts:
                        product_counts[product] = 0
                    product_counts[product] += 1

    # If no orders found
    if total_orders == 0:
        print(f"\033[1;31m{'No orders were found for this user ID.':<40}\033[0m")
        print(f"\033[1;33m{'Redirecting..':<50}\033[0m\n")
        complete_order()
        return

    # Header section
    print(f"\033[1;36m{'=' * 50}\033[0m")
    print(f"\033[1;36m{'CUSTOMER ORDER SUMMARY':^50}\033[0m")
    print(f"\033[1;36m{'=' * 50}\033[0m") #\003 with square brackets is for bolding, :36m is the cyan color,

    # User info
    print(f"\033[1m{'User ID:':<20}\033[0m {userid:>29}")
    print(f"\033[1m{'Total Orders:':<20}\033[0m {total_orders:>29}")
    print(f"\033[1m{'Total Spent:':<20}\033[0m{f'${total_spent:,.2f}':>30}")
    print(f"{'-' * 50}")

    # Product header
    print(f"\033[1;32m{'Product':<30}{'Quantity':>20}\033[0m")
    print(f"{'-' * 50}")

    # Product details
    for product, count in product_counts.items():
        print(f"{product:<30}{count:>20}")

    # Footer
    print(f"\033[1;36m{'=' * 50}\033[0m")
    print(f"\033[1;36m{'Thanks for ordering!':^50}\033[0m")
    print(f"\033[1;36m{'=' * 50}\033[0m")
```

Figure 18. customer summary function

```python
##Lookup_products
def lookup_products(products):

    """
    Kian Diggle

    Look up products is in charge of filtering the scanned list of items. It catches errors like duplicate items (out of stock items since we dont have 2 of each item). It compares the scanned products list with the accepted items and returns a list of the items we want to process for the order
    """

    file = open("products.csv") #opening the scanned products file

    csv_contents = [] #list of the csv file (all accepted items)
    total_contents = [] #list of filtered contents


    for line in file: #get each item from the scanned products
        line_list = line.strip().split(",") #get each element of each prdouct (name, price)
        line_list[1] = float(line_list[1]) #convert the price to an integer (line_list[1] represents the price)

        csv_contents.append(line_list) #append all of line_list (name, price)

    file.close()

    for i in range(len(products)): #for each product in the scanned list
        found_product = False

        for j in range(len(csv_contents) - 1): #for each item in csv_contents (the given list of accepted products)

            if products[i] == csv_contents[j][0]: #if the scanned item name matches a name of any of the accepted items
                total_contents.append(csv_contents[j]) #append the product name and price
                found_product = True

        if found_product == False: #if the product doesnt match with any of the accepted items
            print("Warning: Product not found")

    return total_contents
```

Figure 19. look up products function

```python
##Pack_products


def pack_products(products):
    for i in range(len(products)): #for ezch prodct in the products list

        if products[i] == "Sponge": #check if product is sponge
            sponge()
            pack()
            print("Sponge has been packaged")
        elif products[i] == "Bottle": #check if product is bottle
            bottle()
            pack()
            print("Bottle has been packaged")
        elif products[i] == "Rook": #check if product is rook
            rook()
            pack()
            print("Room has been packaged")
        elif products[i] == "D12": #check if product is D12
            d12()
            pack()
            print("D12 has been packaged")
        elif products[i] == "Bowl": #check if product is bowl
            bowl()
            pack()
            print("Bowl has been packaged")
        elif products[i] == "WitchHat": #check if product is witch hat
            witchhat()
            pack()
            print("Witch hat has been packaged")

#location of each item for the q arm
SPONGE_POSITION = (0.5444417509303867, 0.16565360628151915, 0.1320247864673874)
BOTTLE_POSITION = ((0.5648609134961735, 0.11283654980736692, 0.1393976951869465))
ROOK_POSITION = (0.5725750344624958, 0.04003834675840798, 0.1119978409667301)
D12_POSITION = 0
BOWL_POSITION = 0
WITCHHAT_POSITION = 0
DEPOSIT_POSITION = (0.22789812893019878, -0.2655426667340853, 0.44290507792294694)
time = 0.5
```

Figure 20. Pack products function

```
def pack():
    """
    This function closes the gripper and grabs the object, it then
    moves the object to the packaging station  and returns home
    """
    arm.rotate_gripper(-90)
    sleep(1)
    arm.rotate_gripper(30)
    arm.rotate_gripper(-90)
    sleep(1)
    arm.rotate_gripper(30)
    arm.rotate_gripper(-90)
    sleep(1)
    arm.rotate_gripper(30)
    arm.rotate_gripper(-90)
    sleep(1)
    arm.rotate_gripper(30)
    arm.rotate_gripper(-90)
    sleep(1)
    arm.rotate_shoulder(-30)
    arm.home()
    arm.set_arm_position(DEPOSIT_POSITION)
    arm.rotate_gripper(360)
    arm.home()

def sponge():
    """
    This function moves towards the sponge's location and then it packs it using the pack function, an
    additional thing to note is that it sends the arm above the object then rotates the shoulder so that
    it does not approach the object at a harsh angle
    """
    arm.rotate_gripper(180)
    sleep(time)
    arm.rotate_base(18)
    sleep(time)
    arm.rotate_shoulder(25)
    sleep(time)
    arm.rotate_elbow(-10)
    sleep(time)
    arm.rotate_shoulder(10)
    sleep(time)
    arm.rotate_shoulder(18)
    pack()

def bottle():
    """
    This function is the same as the sponge one, but instead it packs the bottle
    """
    arm.rotate_gripper(360)
    arm.rotate_base(12)
    sleep(time)
    arm.rotate_shoulder(25)
    sleep(time)
    arm.rotate_elbow(-8)
    sleep(time)
    arm.rotate_shoulder(10.5)
    pack()
```

Figure 21. Pack products function

```
def rook():
    """
    This function is the same as the sponge and bottle ones, just for the rook
    """
    arm.rotate_base(5)
    sleep(time)
    arm.rotate_shoulder(30)
    sleep(time)
    arm.rotate_elbow(-6.5)
    sleep(time)
    arm.rotate_shoulder(9.75)
    sleep(time)
    arm.rotate_elbow(-6)
    pack()


def d12():
    """
    This function operates the same as the other ones but for the d12
    """
    for i in range(6):
        arm.rotate_gripper(-120)
        arm.rotate_gripper(30)
    arm.rotate_gripper(300)
    arm.rotate_base(-2)
    sleep(time)
    arm.rotate_shoulder(30)
    sleep(time)
    arm.rotate_elbow(-12.5)
    sleep(time)
    arm.rotate_shoulder(12.5)
    sleep(time)
    arm.rotate_elbow(-3)
    pack()


def witchhat():
    """
    The same functionality as the other ones but for the witchhat !!NOT COMPLETE
    """

    for i in range(6):
        arm.rotate_gripper(-120)
        arm.rotate_gripper(30)
    arm.rotate_gripper(300)
    arm.rotate_base(-8)
    arm.rotate_shoulder(32)
    arm.rotate_elbow(-10)
    for i in range(20):
        arm.rotate_elbow(-2)
    pack()
```

Figure 22. pack products function

```python
def bowl():
    """
    Same thing but just for the bowl !!NOT COMPLETE
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(BOWL_POSITION)
    arm.rotate_shoulder(18)
    pack()

##Complete order
def complete_order(userid, product_list):
    """
    Shayan Siddiqui

    totals an order, applies a random discount, calculates tax,
    prints a professional receipt, and writes the order to orders.csv.

    """

# Subtotal
    subtotal = 0


    for i in range  (len(product_list)):
        price = product_list[i][1] #product_list[i][1] is the product price
        subtotal += price


# Random discount tax and final total
    discount_percent = random.randint(5, 50) #random discount between 5 and 50%
    discount_amount = subtotal * (discount_percent / 100)
    subtotal_after_discount = subtotal - discount_amount

#calculates the tax
    tax_rate = 0.13
    final_total = subtotal_after_discount*(1 + tax_rate)

# Count how many orders already exist
    file = open("orders.csv", "r")

    number_of_orders = 0
    for line in file:
        parts = line.split(",")
        if parts[0] == userid:
            number_of_orders += 1
    file.close()

# Writting the new order to orders.csv
    file = open("orders.csv", "a")

    # build the CSV line
    csv_line = userid + "," + f"{final_total:.2f}"
```

Figure 23. complete order function

```python
    for i in range (len(product_list)):
        product_name = product_list[i][0]
        csv_line = csv_line + "," + product_name


    file.write(csv_line + "\n")
    file.close()

# Professional recipt
    print("\033[1;36m\n============= YOUR ORDER RECEIPT ===============\033[0m")
    print(f"Customer: {userid}")
    print("-------------------------------------------------")
    print("Items Purchased:")

    for i in range (len(product_list)):
        name = product_list[i][0]
        price = product_list[i][1]
        print(f"  - {name:20}                  ${price:6.2f}")


    print("-------------------------------------------------")
    print(f"Subtotal:                                 ${subtotal:6.2f}")
    print(f"Discount ({discount_percent}%):           -${discount_amount:6.2f}")
    print(f"Subtotal After Discount:                  ${subtotal_after_discount:6.2f}")
    print(f"Tax (13%):                                ${tax_rate*subtotal_after_discount:6.2f}")
    print("=================================================")
    print(f"\033[1mFINAL TOTAL:                                 ${final_total:6.2f}\033[0m")
    print("=================================================")

# Order message
    print(f"\nYou have now made {number_of_orders + 1} orders so far.\n")

    return

##Main function
def main():
    """
    the main function brigns together all the functions. It starts by making sure the user is signed in (sign up and authenticate deal with this). It then scans a list of
products. The look up products function is then used to filter the list and return a list of accepted items. Then pack productts is used to go pick up the list fo accepted
items. Complete order then prints a profesionaly formated reciept. If the customer is all doen submiting orders the it calls the customar summary function to give a full
summary of all orders from that customer.
    """

    print("=================================================")
    print("         Welcome to the Warehouse System ")
    print("=================================================\n")

    # User logs in (or signs up)
    userid = authenticate()   # returns userID from the authenticate function
    print(f"\nWelcome, {userid}!\n")

    while True: #keeps looping until the customer no longer wants to complete any orders
        print("\n--- New Order ---")

        #scanning barcodes (returns list of product names)
        scanned_products = scan_barcode() #scan barcode returns a string like "Bowl Sponge D12"
        scanned_products_list =  scanned_products.split()
```

Figure 24. complete order function

```python
def main():
    """
    the main function brigns together all the functions. It starts by making sure the user is signed in (sign up and authenticate deal with this). It t
products. The look up products function is then used to filter the list and return a list of accepted items. Then pack productts is used to go pick up
items. Complete order then prints a profesionaly formated reciept. If the customer is all doen submiting orders the it calls the customar summary funct
summary of all orders from that customer.
    """

    print("================================================")
    print("         Welcome to the Warehouse System ")
    print("================================================\n")

    # User logs in (or signs up)
    userid = authenticate()   # returns userID from the authenticate function
    print(f"\nWelcome, {userid}!\n")

    while True: #keeps looping until the customer no longer wants to complete any orders
        print("\n--- New Order ---")

        #scanning barcodes (returns list of product names)
        scanned_products = scan_barcode() #scan barcode returns a string like "Bowl Sponge D12"
        scanned_products_list =  scanned_products.split()

        #looking up product prices from products.csv
        product_list = lookup_products(scanned_products_list)

        #pack the products with the q-Arm
        pack_products(product_list)

        #compete the order, print receipt + save to orders.csv
        complete_order(userid, product_list)

        # if the user wants to complete another order
        again = input("\nDo you want to process another order? (y/n): ").lower() #make sure the input is lwoercase
        if again != "y": #if its anything other then y
            break #quits the loop as the customer doesnt want to complete any more orders

    # print the customer summary  when user is done
    print("\n\nGenerating customer summary...\n\n")
    customer_summary(userid)

    print("\nThank you for using the Warehouse Ordering System!")


main()
```

Figure 25. Main function

Appendix E: Design Studio Worksheets

Team Worksheet:

MILESTONE 0 (TEAM): COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Habibah Aboueleinin	Aboueh2
Shuja Wazir	Wazirs4
Shayan Siddiqui	Siddm47
Jennifer Desbarats	desbaraj
Kian Diggle	digglek

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

Insert your Team Portrait in the dialog box below

MILESTONE 0 – TEAM CHARTER

Team Tues-52

Project Leads:

Identify team member details (Name and MacID) in the space below.

Role:	Team Member Name:	MacID
Manager	Kian Diggle	digglek
Administrator	Shayan Siddiqui	Siddm47
Coordinator	Jennifer Desbarats	desbaraj
Subject Matter Expert	Habibah Aboueleinin	Aboueh2
Subject Matter Expert	Shuja Wazir	Wazirs4

MILESTONE 0 – PRELIMINARY GANTT CHART (TEAM MANAGER ONLY)

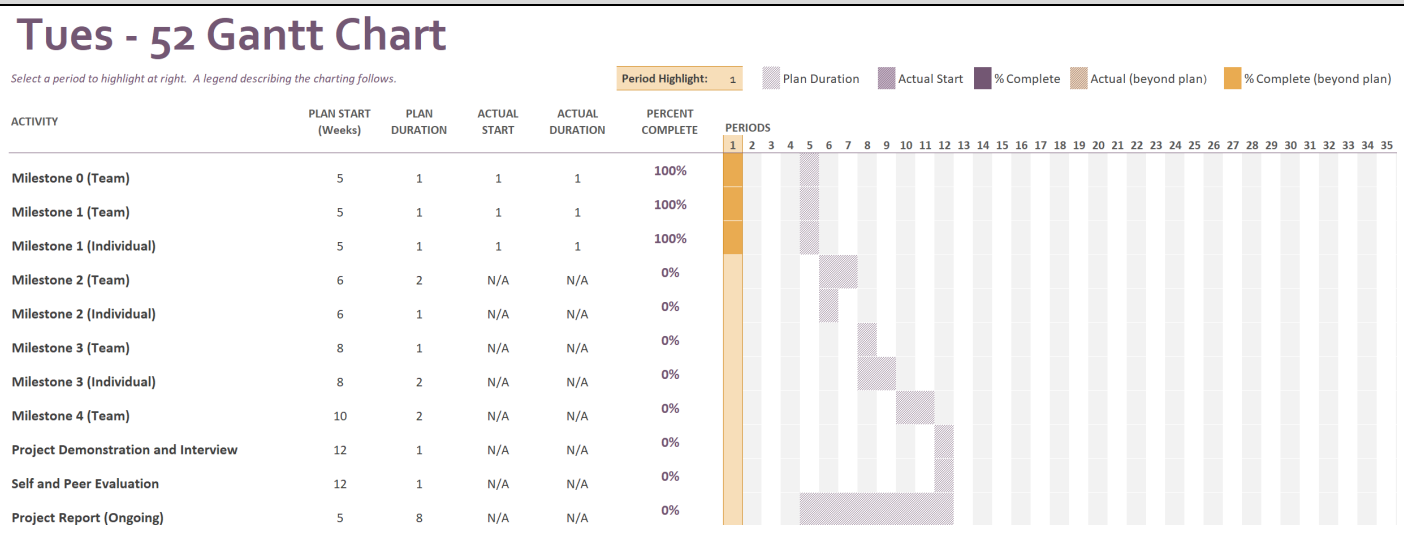
Team ID:

Tues-52

Only the **Project Manager** is completing this section!

Full Name of Team Manager:	**MacID:**
Kian Diggle	digglek

Preliminary Gantt chart



ACTIVITY	PLAN START (Weeks)	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0 (Team)	5	1	1	1	100%
Milestone 1 (Team)	5	1	1	1	100%
Milestone 1 (Individual)	5	1	1	1	100%
Milestone 2 (Team)	6	2	N/A	N/A	0%
Milestone 2 (Individual)	6	1	N/A	N/A	0%
Milestone 3 (Team)	8	1	N/A	N/A	0%
Milestone 3 (Individual)	8	2	N/A	N/A	0%
Milestone 4 (Team)	10	2	N/A	N/A	0%
Project Demonstration and Interview	12	1	N/A	N/A	0%
Self and Peer Evaluation	12	1	N/A	N/A	0%
Project Report (Ongoing)	5	8	N/A	N/A	0%

MILESTONE 1 (TEAM): COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Jennifer Desbarats	desbaraj
Shayan Siddiqui	Siddm47
Kian Diggle	Digglek
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

MILESTONE 1 (STAGE 1) – PROBLEM STATEMENT, FUNCTIONS, OBJECTIVES AND CONSTRAINTS

Team ID: Tues-52

Write out your initial problem statement in the box below.

Initial Problem Statement
In order to assist in the facilitation of global warehouses, objects must be retrieved from an area and deposited in a parcel. This will not only ease the burden on warehouse workers, but improve the quality of global warehouses. The task requires us to pick up different objects of varying difficulty, and return them to a drop-off parcel. Each object has a different shape, size, and rigidity, and we must successfully pick up and deposit each object in order to successfully complete the full task.

List out your functions, objectives and constraints in the box below.

Functions	**Objectives**	**Constraints**
- Grips securely onto object - Moves to accommodate shape and size - Locates and retrieves correct object each time - Sorting pieces to drop off	- Should have a strong grip - Should have good mobility - Should be versatile to pick up different shapes - Should be able to reach all required objects - Should complete task quickly/efficiently - Should be able to complete the task in its entirety - Should be environmentally sustainable, minimal waste	- Can not require excess material/weight - Must be easily able to fabricate within the timeline - Must operate within safe conditions - Reduce strain on warehouse staff - Handle objects with care when retrieving and depositing

MILESTONE 1 (STAGE 2) – TA CHECK-IN

During and after your TA check-in document their feedback in the box below.

Feedback
Problem statement - Good - Lacking initial problem at hand (fictitious prompt) - Warehouse worker conditions and lack of efficiency in current mechanisms Functions - Good don't need change Objectives - Good - Add one about how system should be efficient Constraints - Move efficiency constraint to objectives - Should reduce strain on staff in warehouse - Must have safe operations

List out action items to address any of your team's concerns and your TA's feedback

Action Items
- Adding initial problem based on prompt (sustainable, improving warehouse conditions) - focus on sustainability (adapt objectives and constraints to this focus) - Adapt constraints to keep warehouse worker in mind

MILESTONE 2 (TEAM) – COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Jennifer Desbarats	desbaraj
Kian Diggle	Digglek
Shayan Siddiqui	Siddm47
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

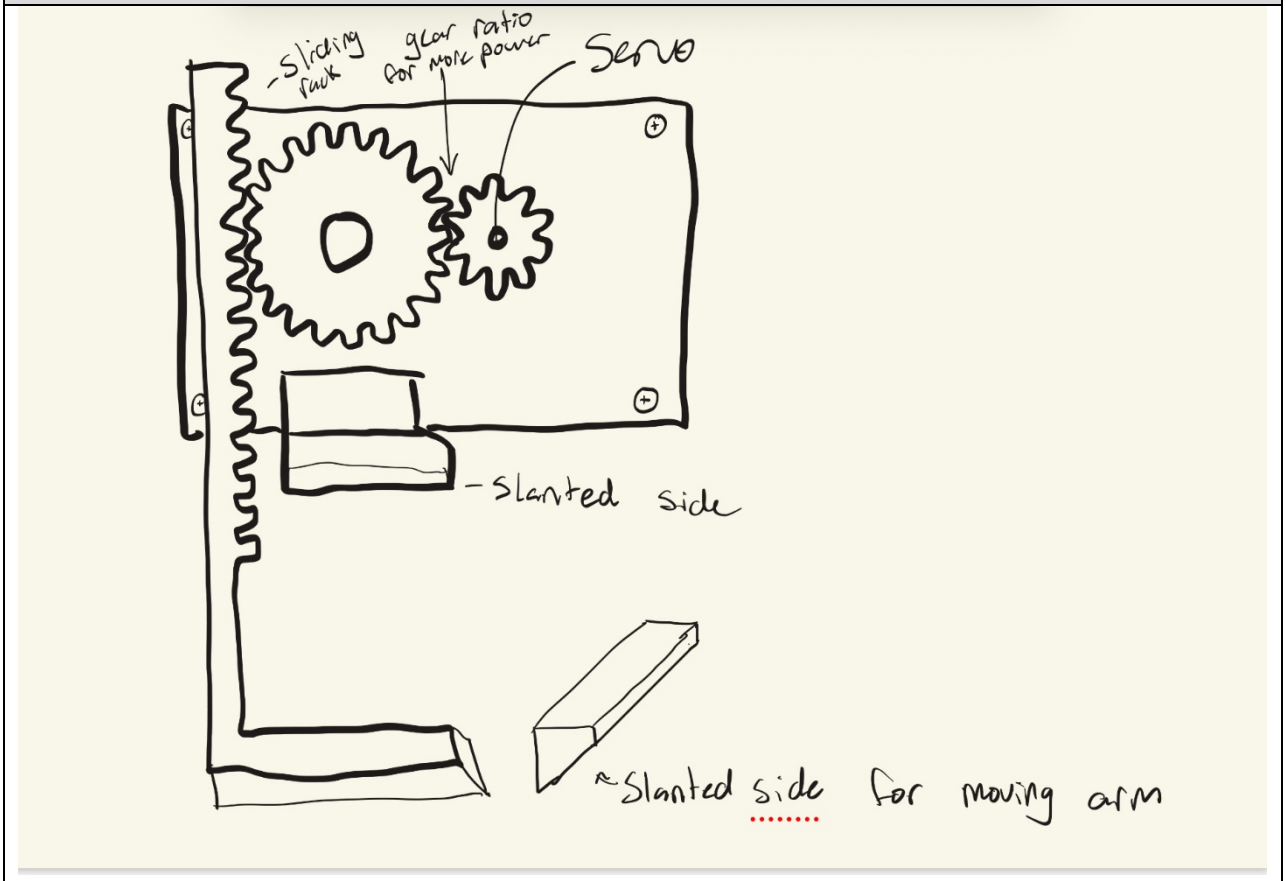
Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

MILESTONE 2 (STAGE 2) – MECHANISM DECISION AND CAD MODEL

Team ID: Tues-52

Attach your teams chosen finalized mechanism concept sketch in the following box

Finalized Concept Sketch



Justify why your team chose this design

Justification

- Simplicity, very few moving parts and can be easily created/modified
- Versatile, has both flat and angled edges to pick up various shapes
- Rotational motion from gears is converted to linear motion in the direction needed to retrieve and grip objects

Modifications:

Base piece:

- Create holes for new fastener positions
- Create an angled arm attached to the base

Arms:

- Adapt arm shape to more effectively pick up all shapes (curve them and slant them)

Rack and sliding piece

- Move slider out of further from the middle of the base, so that the rack clears the side of the end effector platform

Gears:

- Increase the widths of both gears and all overall features
- Increase the amount of teeth on the second gear, to account for the rack being moved further

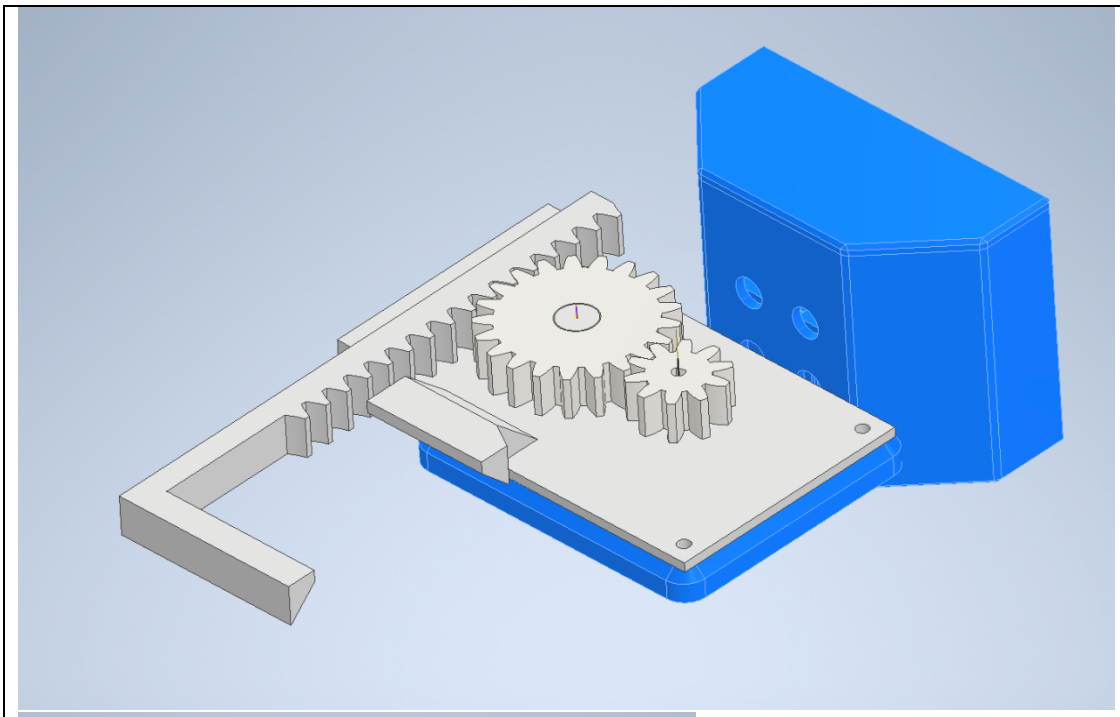
Overall:

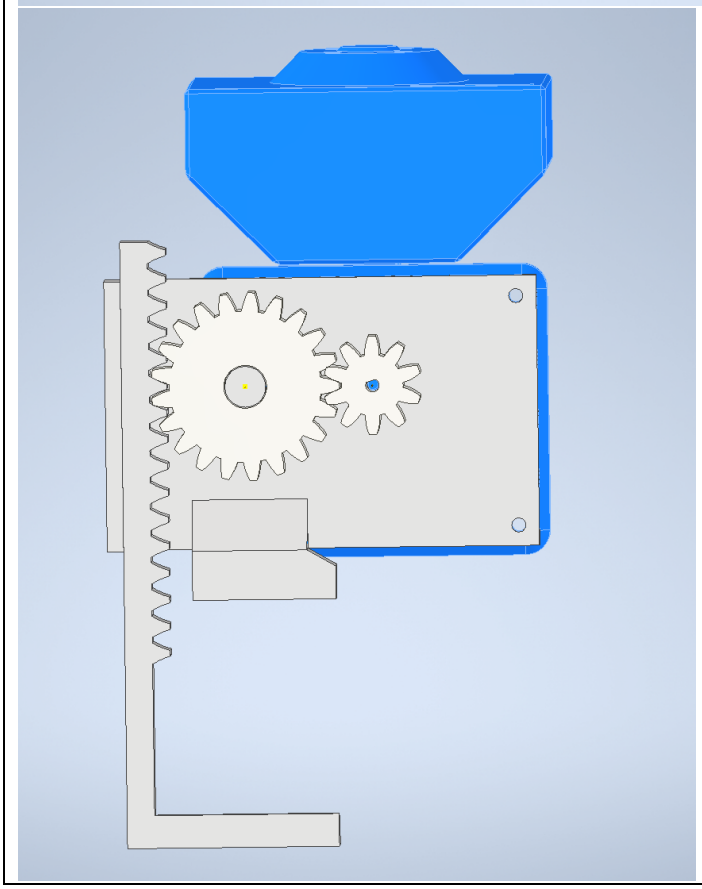
- Add 0.5mm – 1mm gap between all moving parts

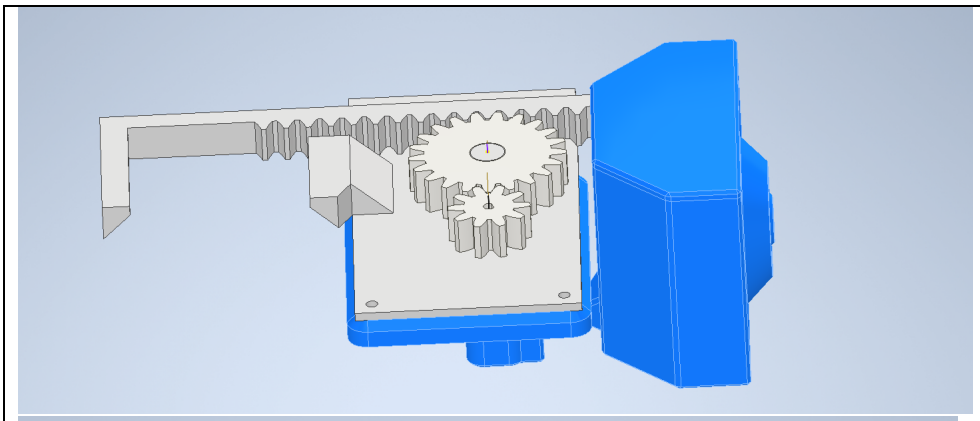
Attach multiple screenshots of different views of your CAD model to the following box

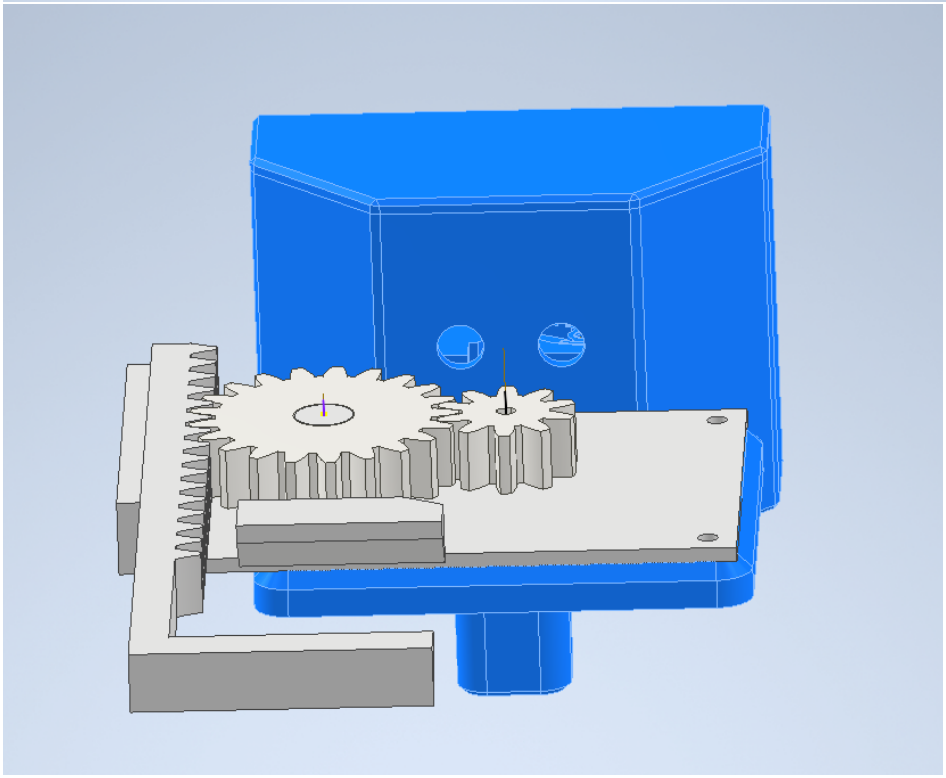
Stronger grip, done through the angled arms and moveable/fixed arm. Versatility, as the arms have both angled and flat surfaces on them. Environmentally friendly, minimal parts needed.

CAD Model Screenshots









Select one or more objectives for your design and explain how you plan to further optimize it for each.

MILESTONE 2 (STAGE 3) –TA CHECK-IN

During and after your TA check-in document their feedback in the box below.

Feedback
- Keep 0.5 – 1 mm gap to account for friction between the pieces - Increase the thickness of the arms to improve the sturdiness

List out action items to address any of your team's concerns and your TA's feedback

Action Items
Base piece: - Create holes for new fastener positions - Create an angled arm attached to the base - Arms: - Adapt arm shape to more effectively pick up all shapes (curve them and slant them) **Rack and sliding piece** - Move slider out of further from the middle of the base, so that the rack clears the side of the end effector platform - Gears: - Increase the widths of both gears and all overall features - Increase the number of teeth on the second gear, to account for the rack being moved further **Overall:** Add 0.5mm – 1mm gap between all moving parts ** **All the Feedback Has already been implemented into the design shown above****

WORK PERIOD: FABRICATION TIME – COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Kian Diggle	digglek
Jennifer Desbarats	desbaraj
Shayan Siddiqui	Siddm47
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

WORK PERIOD: FABRICATION TIME (STAGE 3) – TA CHECK-IN

Team ID: Tues-52

During and after your TA check-in document their feedback in the box below.

Feedback
Looks good. - Some adjustments might need to be made based on the q-arm constraints - The fitting of the rack and gear requires testing after printing is done

List out action items to address any of your team's concerns and your TA's feedback

Action Items
- Print the current CAD design - Test out the claw - Get started on Q-Arm code - Think of possible improvements to the design

FABRICATION APPROVAL

Design Meets Design Objectives

→ Mechanism simulates motion of picking up objects using the servo motor

→ Mechanism and servo motor are properly toleranced according to the engineering drawing

Assembly model is complete and aesthetic, properly grounded and has no interference or errors

Design intentionally minimizes materials; the entire mechanism including fasteners weighs less than 100 grams.

Only non-flat components are 3D printed

☐ ALL mechanism features are 1mm or more

→ Not only do features need to be 1mm or greater, but spaces between them as well

→ Features between 2mm and 4mm are appropriately sized and will not compromise the fabricated design

☐ **APPROVED FOR PRINTING**

MILESTONE 3 (TEAM) – COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Kian Diggle	digglek
Jennifer Desbarats	desbaraj
Shayan Siddiqui	Siddm47
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

MILESTONE 3 (STAGE 1) – FUNCTION ASSIGNMENT

Team ID: **Tues-52**

Using the table below assign each function to one member of your team.

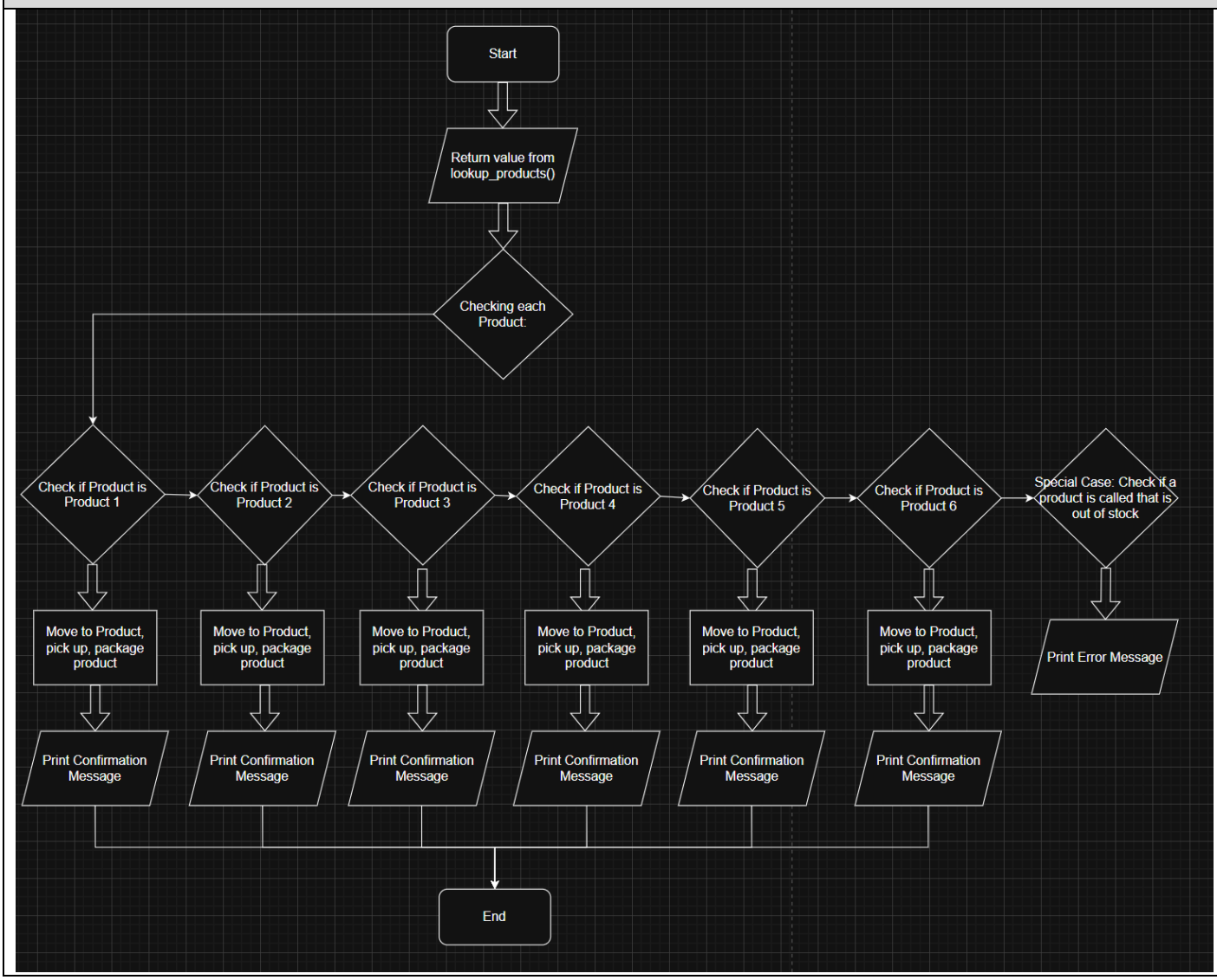
Function	Full Name	MacID
sign_up()	Shuja Wazir	Wazirs4
authenticate()	Jennifer Desbarats	desbaraj
lookup_products()	Kian Diggle	digglek
complete_order()	Shayan Siddiqui	Siddm47
customer_summary()	Habibah Aboueleinin	Aboueh2

MILESTONE 3 (STAGE 2) – FLOWCHART

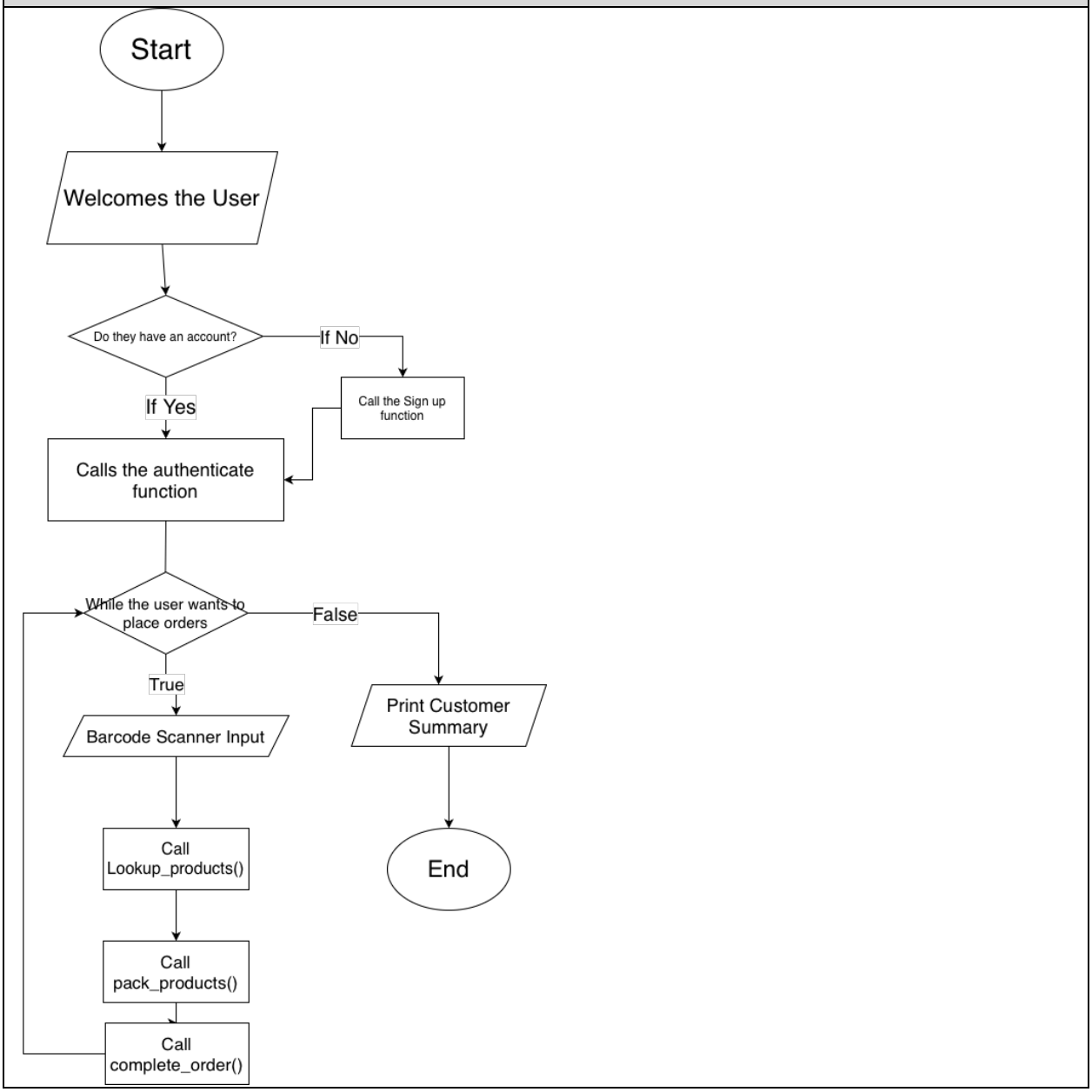
Action Items
Pack products: - Modify terminology for checking products (no python terminology) - Add arrows for checking each product and between the print confirmation message and end Main: - Edit to call specific function names

Insert images of flowcharts in the boxes below for the two group functions: pack_products() and main().

pack_products()



main()



MILESTONE 3 (STAGE 3) –TA CHECK-IN

During and after your TA check-in document their feedback in the box below.

Feedback

For pack_products():

- Do not use Python terminology in flowchart
- Add arrows connecting the beginning of the loop to each stage of the iteration
- Add arrows connecting the output statements to the end

For main():

- Use function names
- Put complete order inside the while loop

** Feedback already implemented

List out action items to address any of your team's concerns and your TA's feedback

WORK PERIOD: CODE YOUR FUNCTION – COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	MacID:
Kian Diggle	digglek
Jennifer Desbarats	desbaraj
Shayan Siddiqui	Siddm47
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

WORK PERIOD: CODE YOUR FUNCTION (STAGE 1) – TA CHECK-IN

Team ID: Tues-52

During and after your TA check-in document their feedback in the box below.

Feedback
- Looks Good

List out action items to address any of your team's concerns and your TA's feedback

Action Items
- Finish Individual code functions - Integrate individual functions together - Continue working on team functions

MILESTONE 4 (TEAM) – COVER PAGE

Team ID: Tues-52

Please list full names and MacIDs of all *present* Team Members

Full Name:	**MacID:**
Kian Diggle	digglek
Jennifer Desbarats	desbaraj
Shayan Siddiqui	Siddm47
Shuja Wazir	Wazirs4
Habibah Aboueleinin	Aboueh2

Any student that is ***not*** present for Design Studio will not be given credit for completion of the worksheet and may be subject to a deduction to their P-1 grade. See course outline for details.

MILESTONE 4 (STAGE 1) – Q-ARM CODE

Insert screenshots of your Q-Arm code to the box below.

Q-Arm Code Screenshots

```python
# !/usr/bin/env python3
# coding: utf-8
# ------------------------------------------------------------------------------


import sys

sys.path.append("../")

from time import sleep
from Common.qarm_interface_wrapper import *

GRIPPER_IMPLEMENTATION = 1
arm = QArmInterface(GRIPPER_IMPLEMENTATION)
scan_barcode = BarcodeScanner.scan_barcode



# ------------------------------------------------------------------------------
# STUDENT CODE BEGINS
# ------------------------------------------------------------------------------

#GRIPPER MAX OPEN (~ 180 deg)
#SHOULDER MAX (~ 50 deg)
#ELBOW MAX (~ -50 deg)

##These are constant values of the positions of each object
SPONGE_POSITION = (0.5588460498271793, 0.17567335117809524, 0.19145975061150977)
BOTTLE_POSITION = ((0.5417731998260051, 0.10908926028689474, 0.3151814295407535))
SPONGE_POSITION = (0.5588460498271793, 0.17567335117809524, 0.19145975061150977)
BOTTLE_POSITION = ((0.5417731998260051, 0.10908926028689474, 0.3151814295407535))
ROOK_POSITION = (0.5725750344624958, 0.04003834675840798, 0.11199784809667301)
D12_POSITION = 0
BOWL_POSITION = 0
WITCHHAT_POSITION = 0
DEPOSIT_POSITION = 0

# STILL NEED TO DETERMINE THE POSITIONS OF THE OTHER OBJECTS


def pack():
    """
    This function closes the gripper and grabs the object, it then
    moves the object to the packaging station and returns home
    """
    arm.rotate_gripper(-180)
    sleep(1)
    arm.rotate_shoulder(-20)
    arm.home()
    arm.set_arm_position(DEPOSIT_POSITION)
    arm.rotate_gripper(180)
    arm.home()

def sponge():
    """
    This function moves towards the sponge's location and then it packs it using the pack function, an
    additional thing to note is that it sends the arm above the object then rotates the shoulder so that
    it does not approach the object at a harsh angle
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(SPONGE_POSITION)
    arm.rotate_shoulder(18)
    pack()
```

```
def sponge():
    """
    This function moves towards the sponge's location and then it packs it using the pack function, an
    additional thing to note is that it sends the arm above the object then rotates the shoulder so that
    it does not approach the object at a harsh angle
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(SPONGE_POSITION)
    arm.rotate_shoulder(18)
    pack()


def bottle():
    """
    This function is the same as the sponge one, but instead it packs the bottle
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(BOTTLE_POSITION)
    arm.rotate_shoulder(18)
    pack()

def rook():
    """
    This function is the same as the sponge and bottle ones, just for the rook
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(ROOK_POSITION)
    arm.rotate_shoulder(18)
    pack()

def d12():
    """
    This function operates the same as the other ones but for the d12
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(D12_POSITION)
    arm.rotate_shoulder(18)
    pack()

def witchhat():
    """
    The same functionality as the other ones but for the witchhat
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(WITCHHAT_POSITION)
    arm.rotate_shoulder(18)
    pack()

def bowl():
    """
    Same thing but just for the bowl
    """
    arm.rotate_gripper(180)
    arm.set_arm_position(BOWL_POSITION)
    arm.rotate_shoulder(18)
    pack()









# ----------------------------------------------------------------------------
# STUDENT CODE ENDS
# ----------------------------------------------------------------------------
```

MILESTONE 4 (STAGE 3) – TA CHECK-IN

During and after your TA check-in document their feedback in the box below.

Feedback
- Make the arm on the rack longer.

List out action items to address any of your team's concerns and your TA's feedback

Action Items
- Reprint fixed mechanism with longer arm. - Complete Q-arm code for each object. - Fix gears to prevent getting stuck.

Milestone 1 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

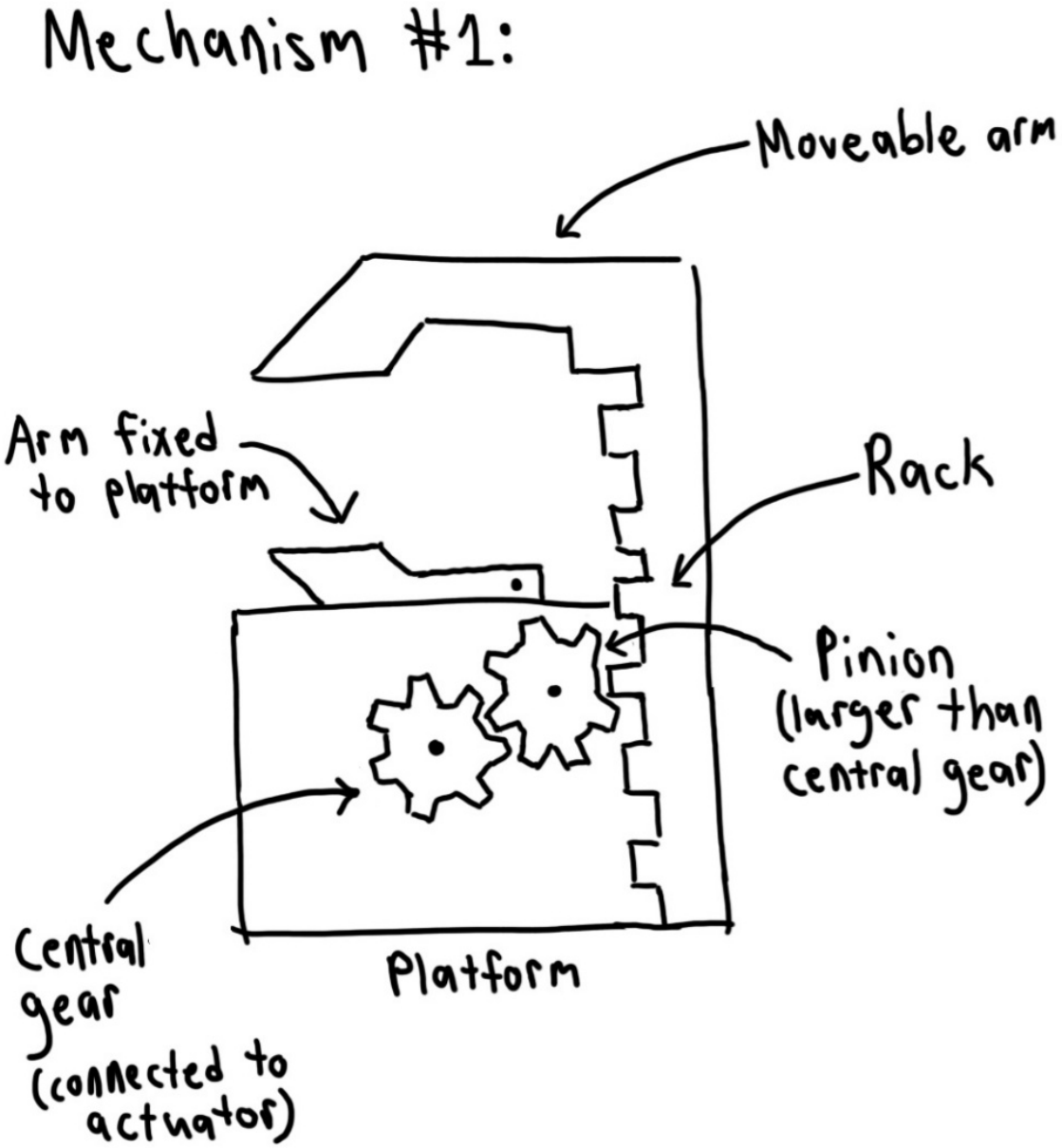
Full Name:	MacID:
Kian Diggle	digglek

Milestone 1 (Stage 2) – Mechanism Concept sketches

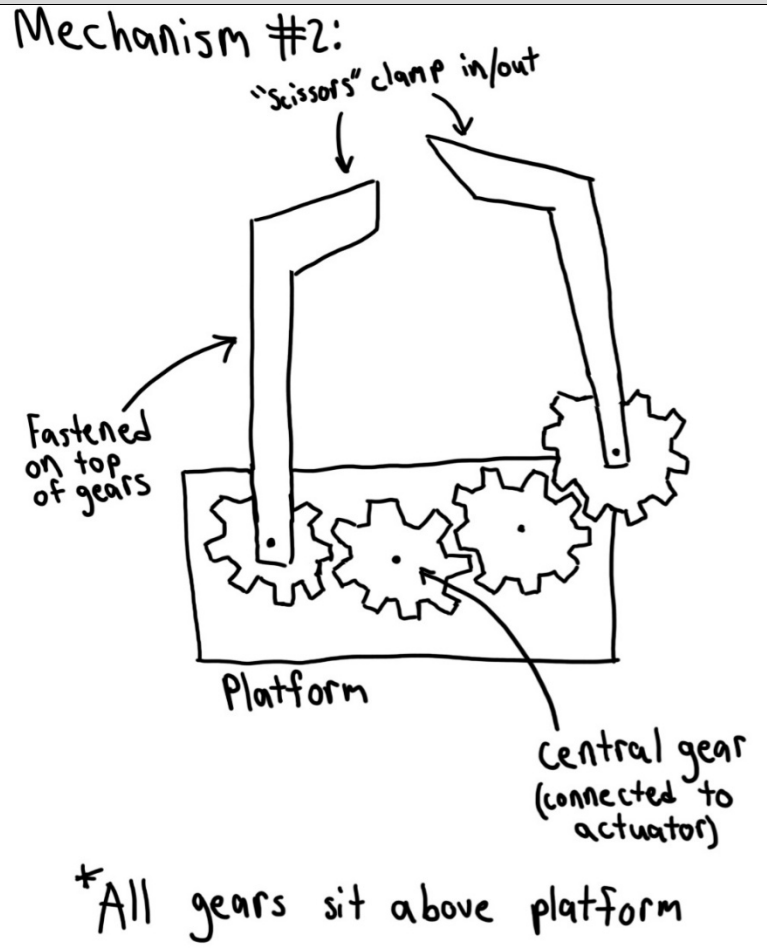
Team ID: Tues-52

Insert images of your initial mechanism sketches in the box below.

Sketch #1



Sketch #2



Select one or more objectives for your design and explain how you plan to optimize it for each.

Justification

- Strong grip, used in Sketch #1 by the horizontally fixed arm, pushing against the moveable arm on the rack
- Strong grip, used in Sketch #2 by the "scissors/clamp," where the object would be gripped well using the two arms
- Mobility, used in Sketch #1 through the moveable rack sliding against the fixed arm
- Mobility, used in Sketch #2 through the two arms, which sit above the gears and are controlled through the central gear
- Will make each design as lighweight as possible, to reduce waste and optimize
- Reduce the weight of the rack and central gears, as well as making the clamps extremely light

Milestone 2 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

Full Name:	**MacID:**
Kian Diggle	digglek

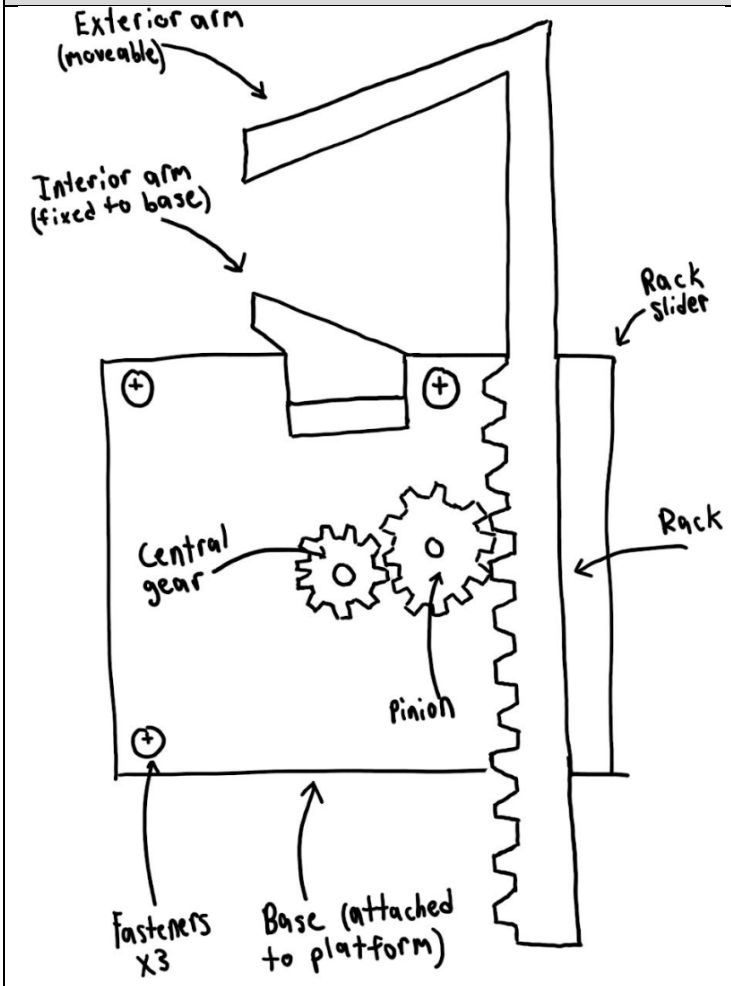
Milestone 2 (Stage 1) – Finalized Concept Sketches and CAD Model worksheet

Team ID: Tues-52

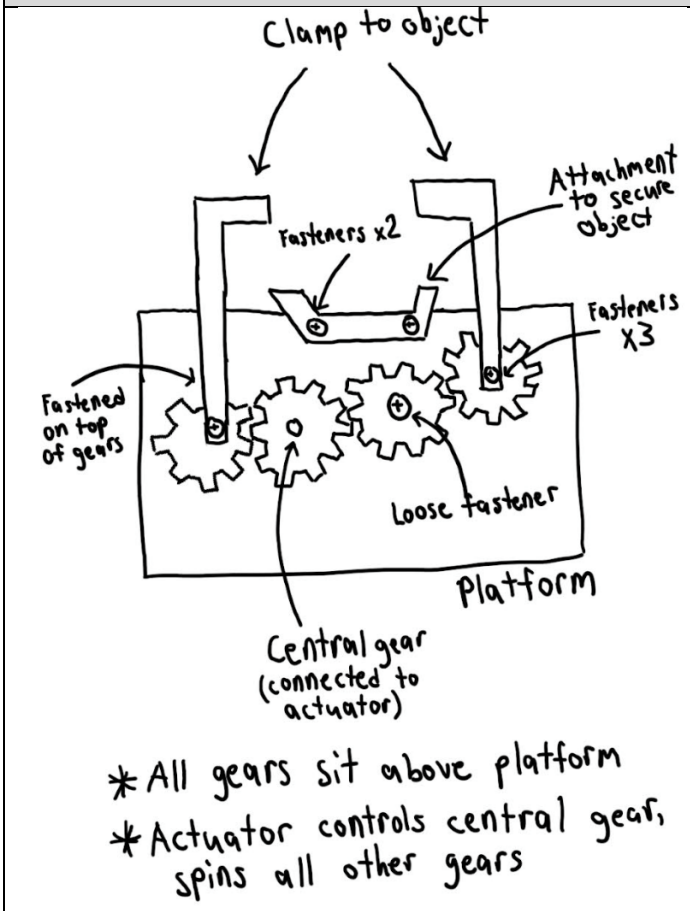
This is an individual deliverable and should be completed by each team member **prior** to Design Studio.

Attach images to the following three boxes of your mechanism sketch concepts and ideas.

Refined Sketch #1

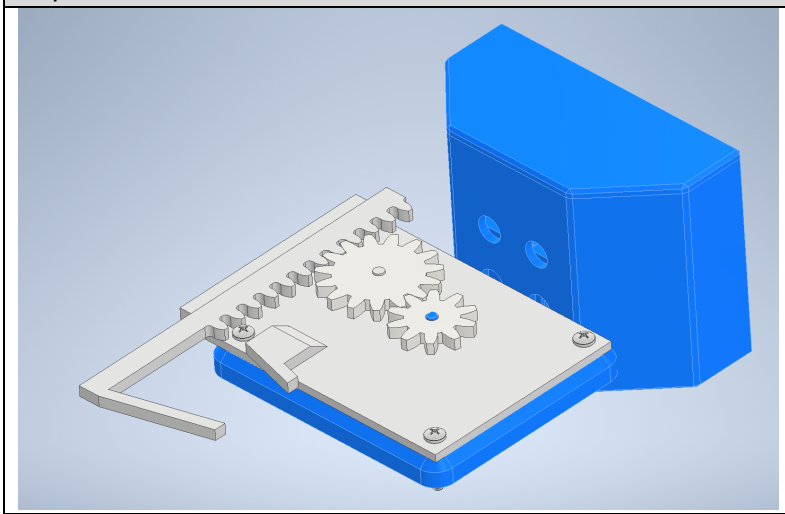


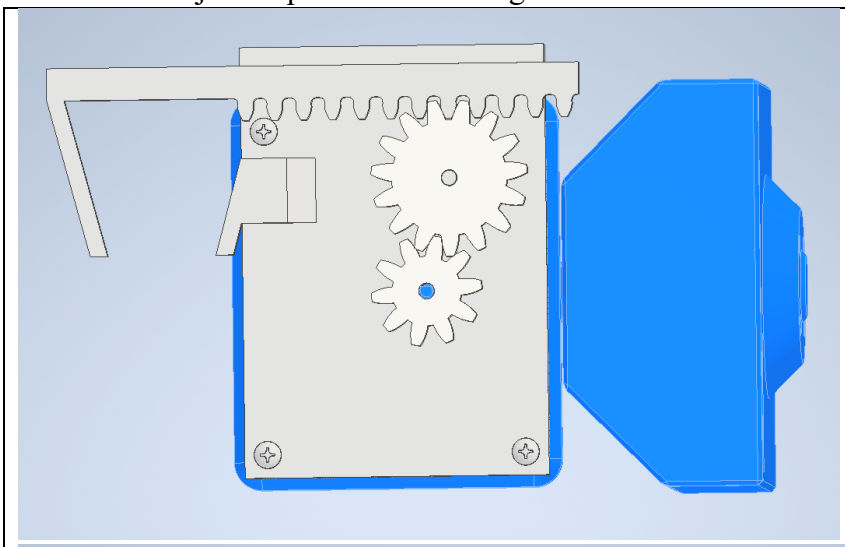
Refined Sketch #2

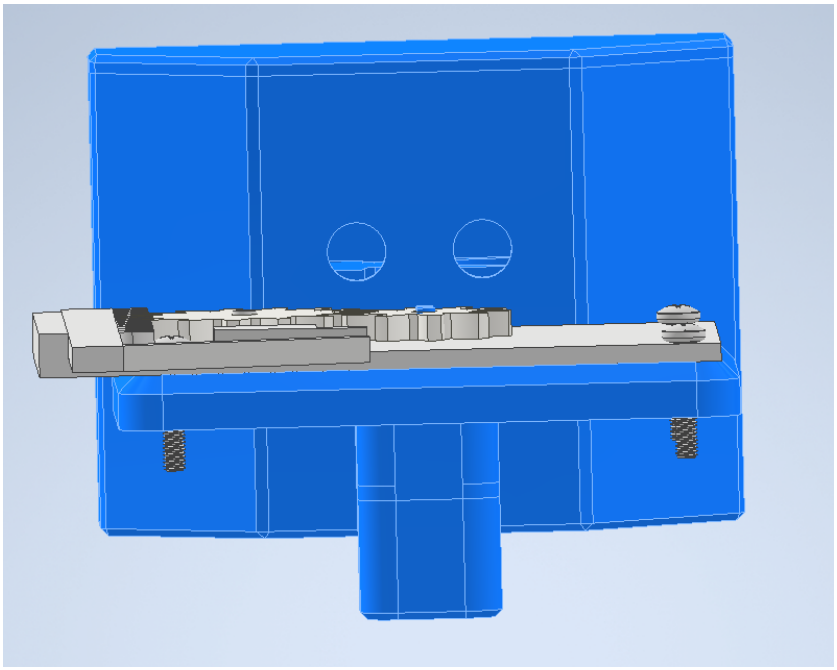


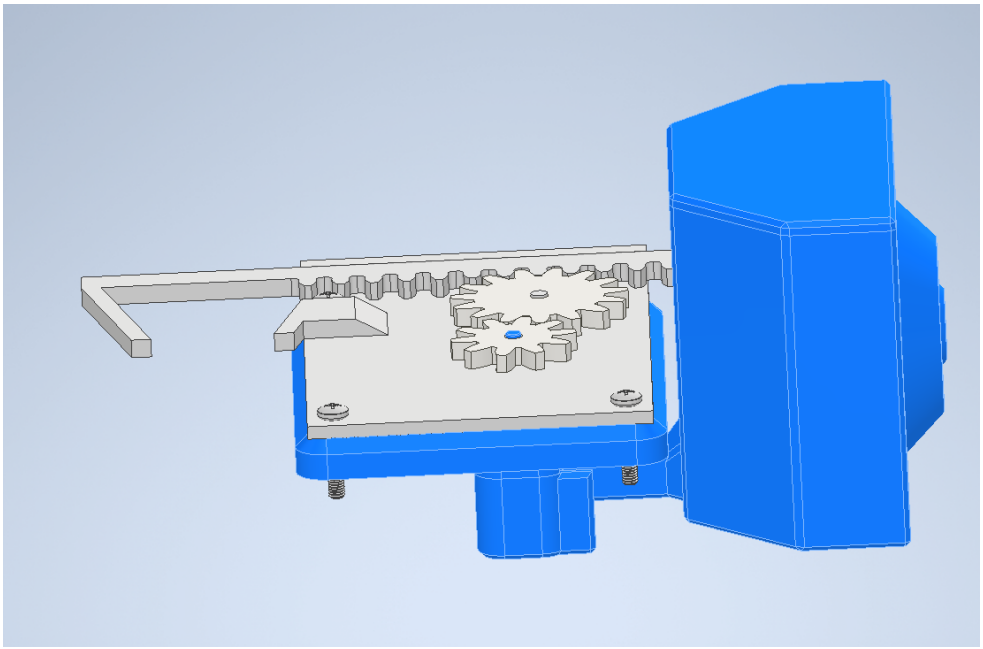
CAD Model Screenshots

Include screenshots of the top, front, right and isometric views of your CAD model. All design features of your CAD model should be visible.









Select one or more objectives for your design and explain how you plan to further optimize it for each.

Justification
-Stronger grip/versatility, which can be done through slanted arm pieces. This will predominantly aid in picking up objects 4 and 5, which have slanted main bodies -Make the rack longer, to be able to reach all objects and accommodate all widths -Curved shape for the moveable and fixed arm -Extend the base to accommodate for a longer rack

Milestone 3 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

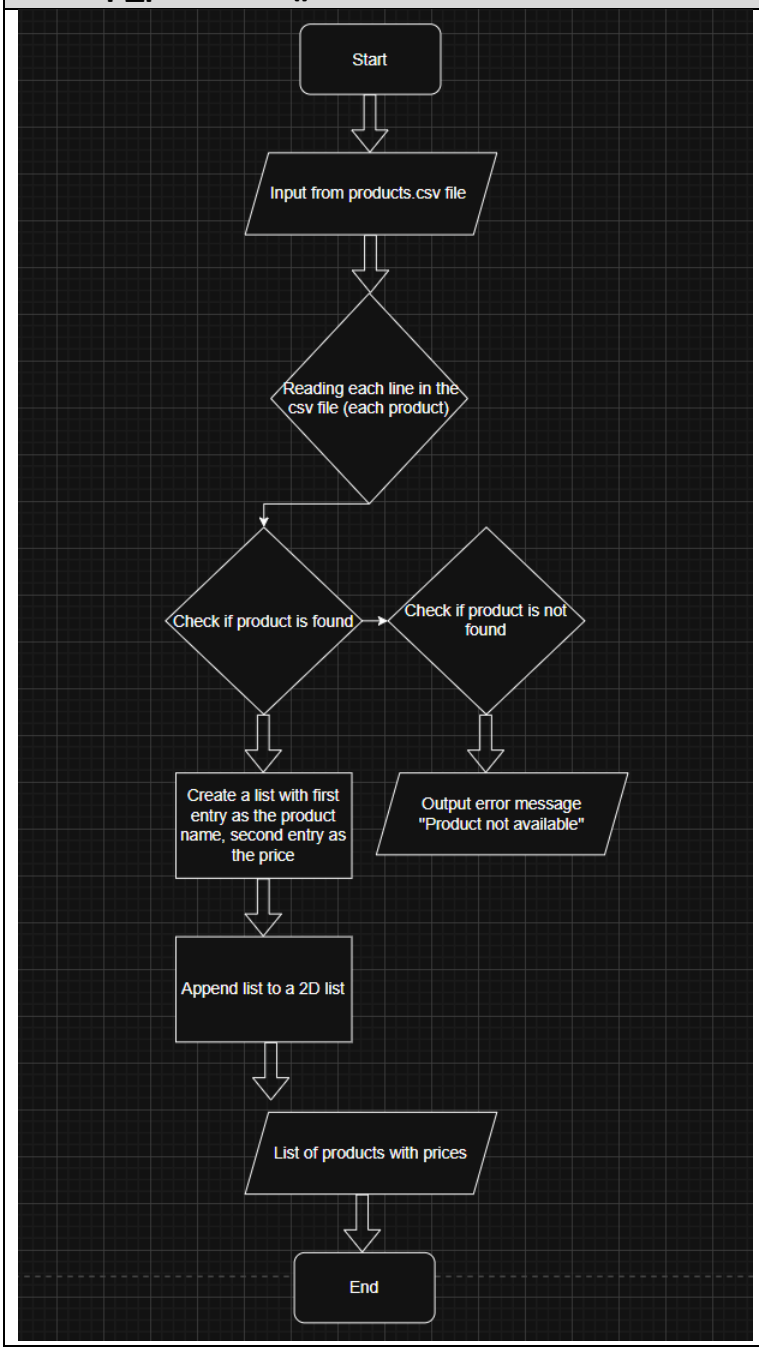
Full Name:	**MacID:**
Kian Diggle	digglek

Milestone 3 (Stage 2) – Flowchart

Team ID: Tues-52

Insert an image of flowchart in the box for your assigned function. Also change the name of the function to match your assigned function.

lookup_products ()



Milestone 1 (Individual) – Cover Page

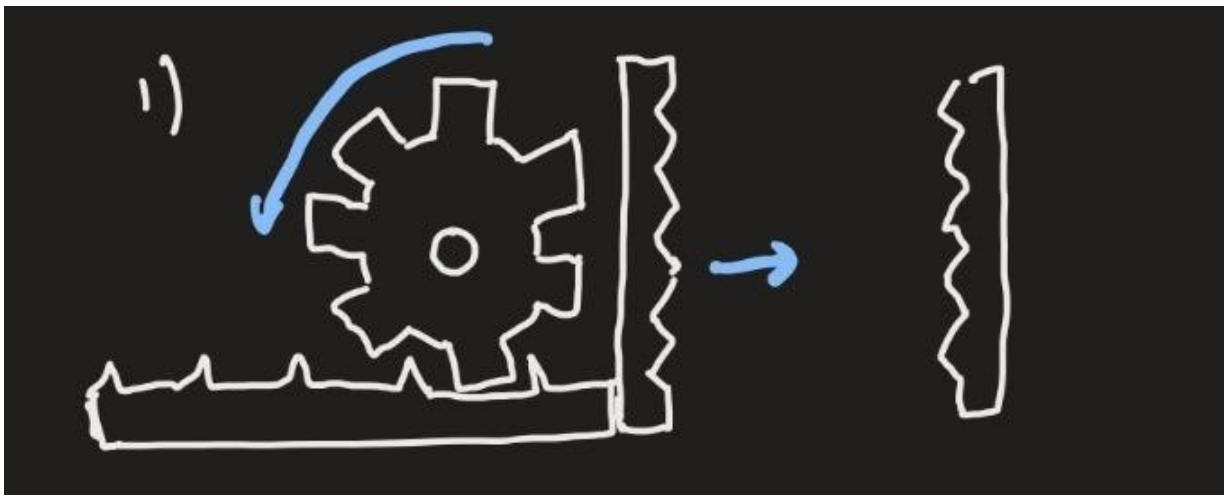
Team ID: Tues-52

Please list full name and MacID.

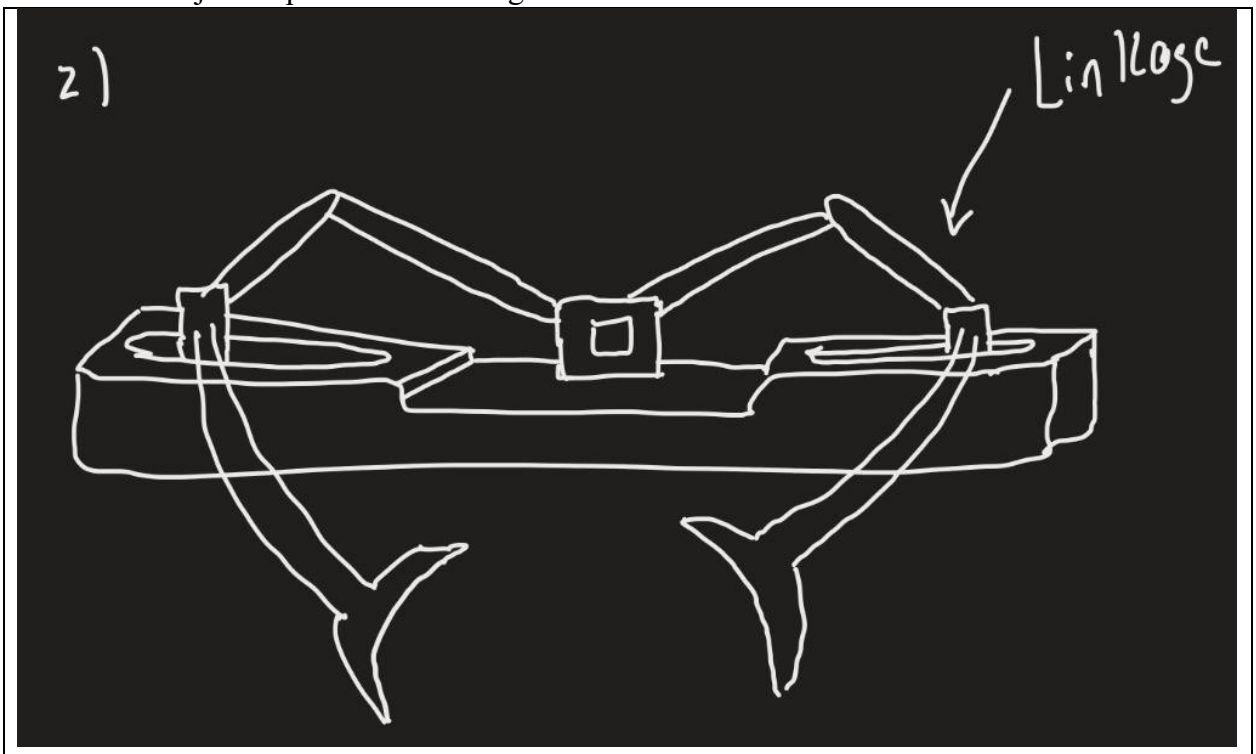
Full Name:	MacID:
Shuja Wazir	Wazirs4

Insert images of your initial mechanism sketches in the box below.

Sketch #1



Sketch #2



Select one or more objectives for your design and explain how you plan to optimize it for each.

Justification
With design number 1, as the gear rotates the rack and pinion closes until the spikes meet creating a pinching force that grips onto the object and holds it tight. An important thing to consider is that only one side moves and the other one stays stationary, in this example the leftmost side is the one that is moving, and the rightmost side is stationary.

Justification
With design number 2, the servo is connected to the piece in the middle and as it rotates it moves the linkages closer together so that the arms at the end meet up but not fully clamped, the idea with this one is for more circular objects so that they may be picked up at the edges whereas the first design may not able to pick up circular objects as efficiently.

Milestone 2 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

Full Name:	**MacID:**
Shuja Wazir	wazirs4

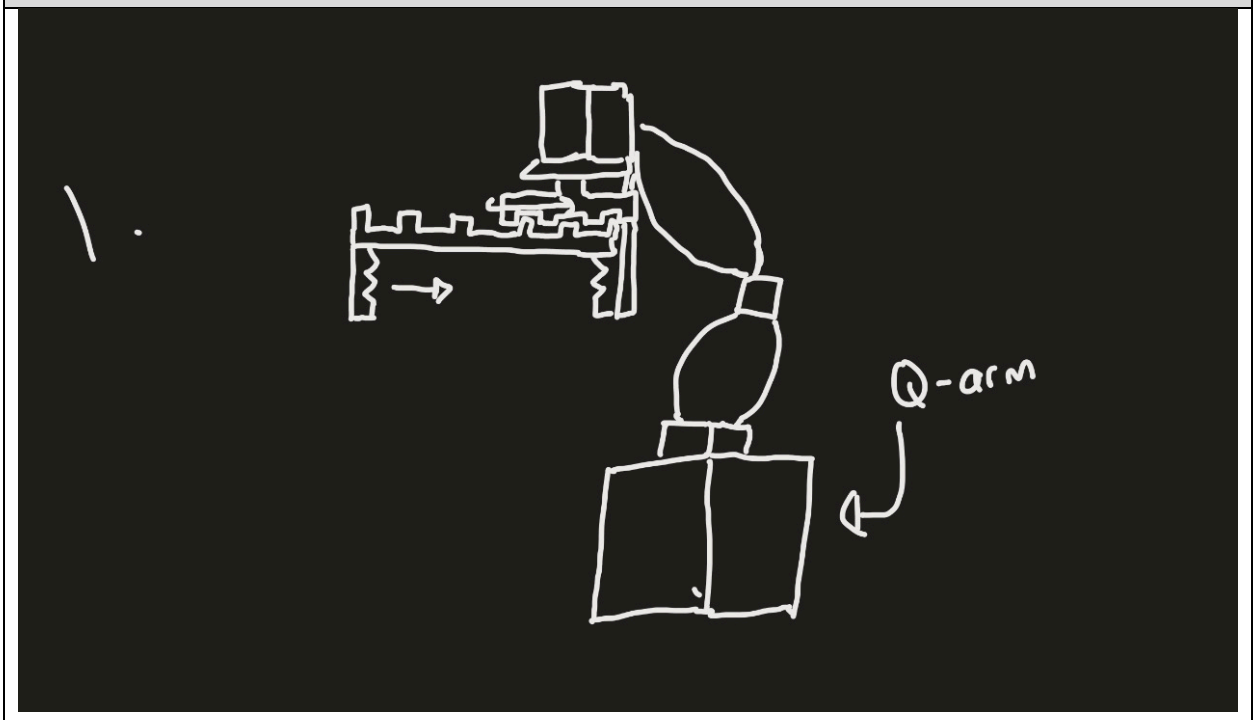
Milestone 2 (Stage 1) – Finalized Concept Sketches and CAD Model worksheet

Team ID: Tues-52

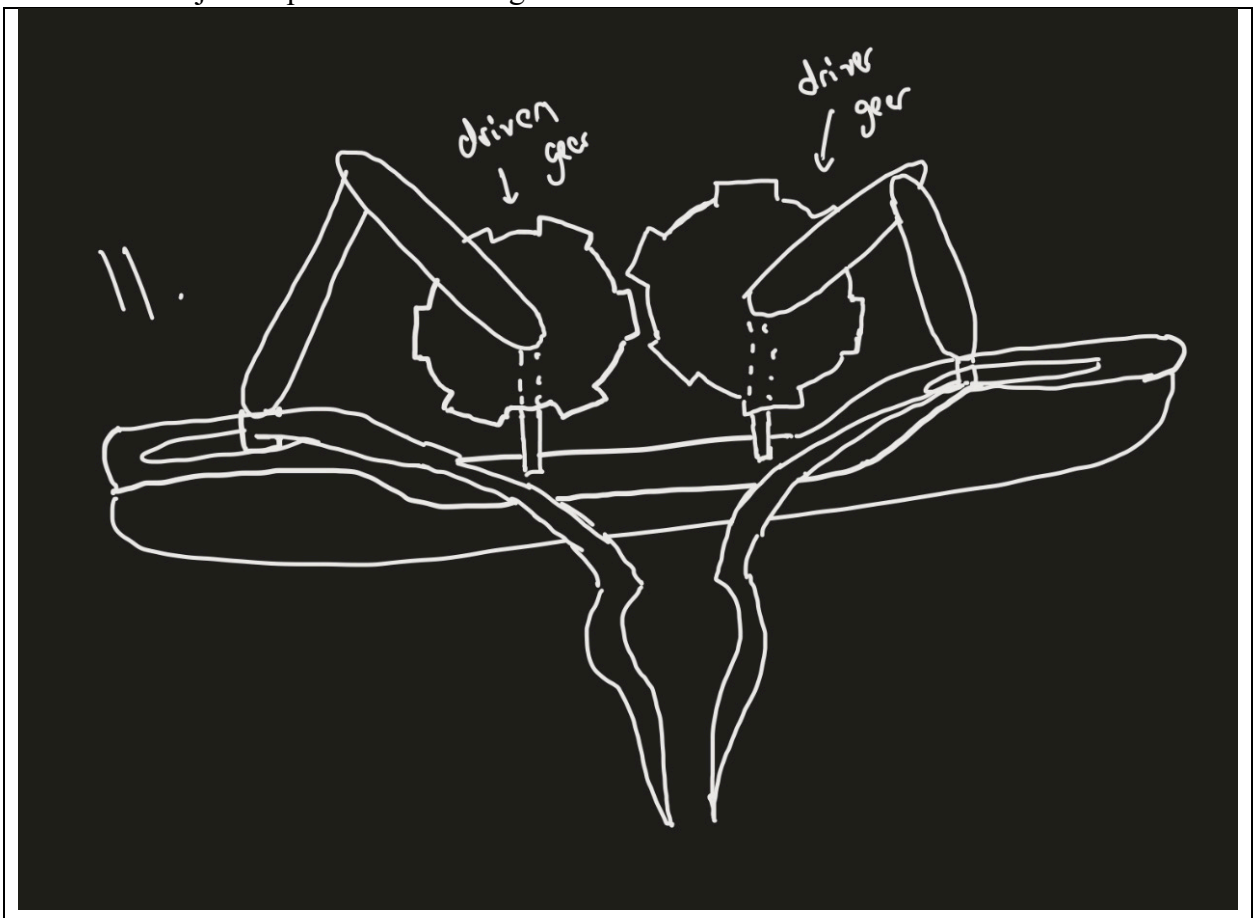
This is an individual deliverable and should be completed by each team member **prior** to Design Studio.

Attach images to the following three boxes of your mechanism sketch concepts and ideas.

Refined Sketch #1

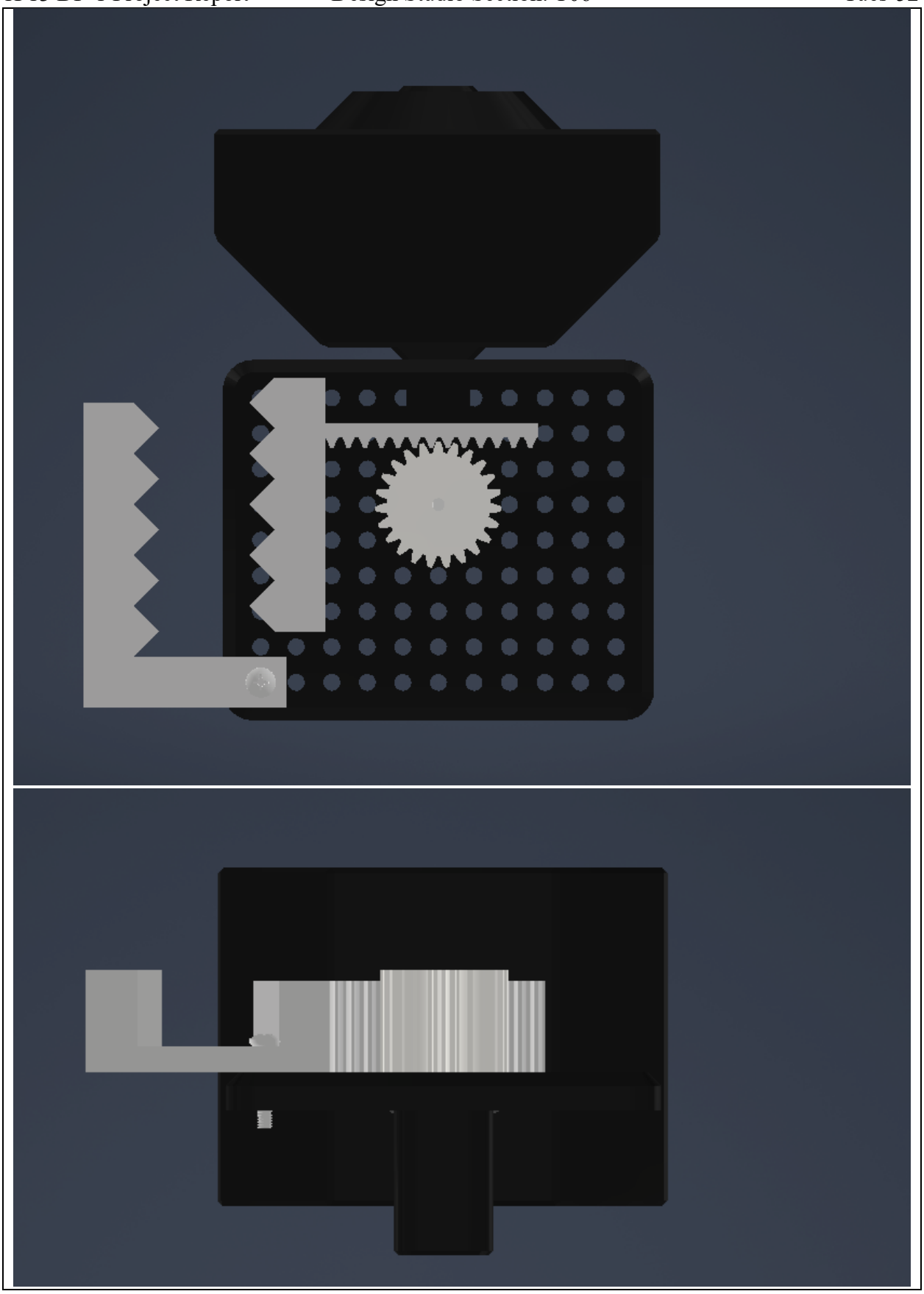


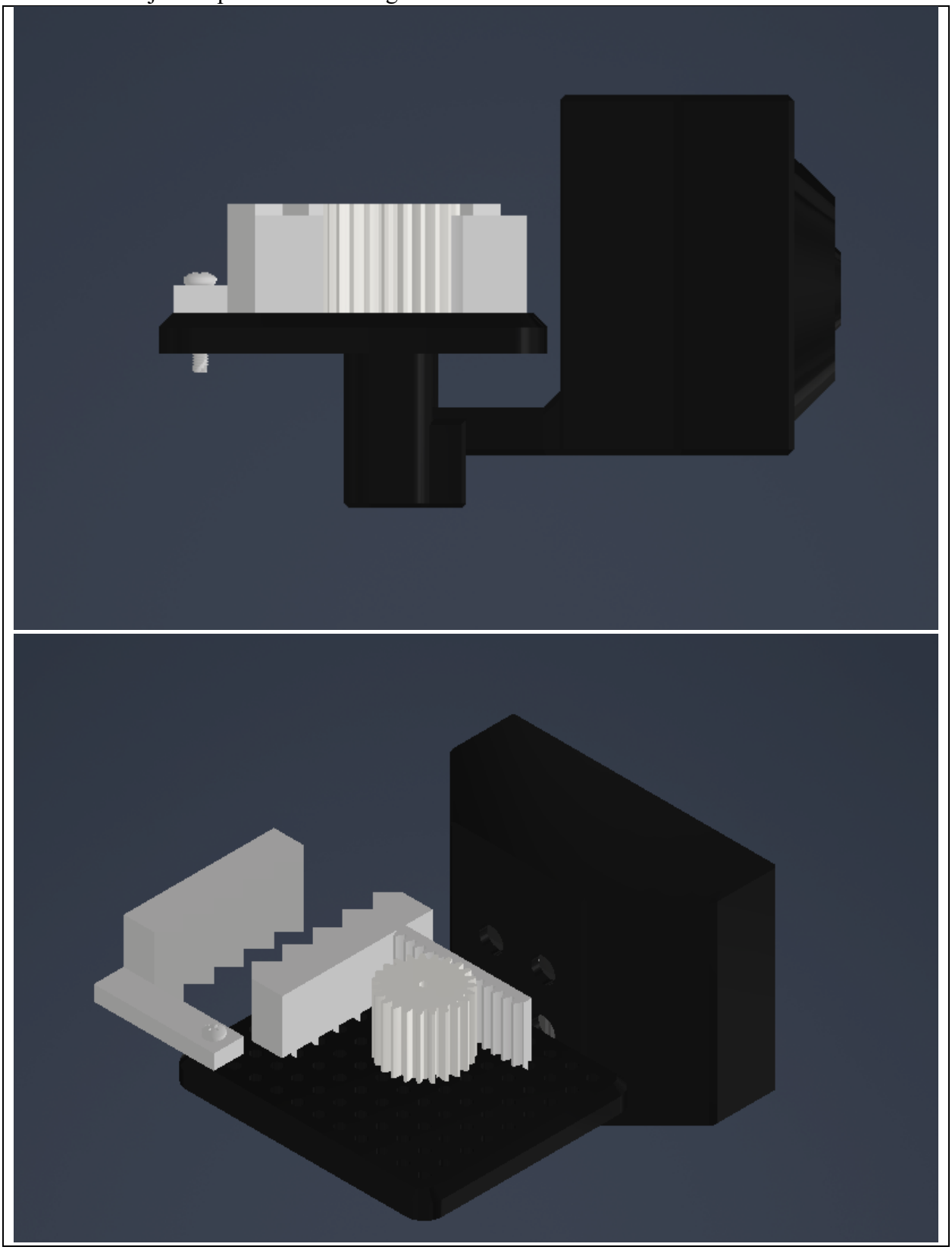
Refined Sketch #2



CAD Model Screenshots

Include screenshots of the top, front, right and isometric views of your CAD model. All design features of your CAD model should be visible.





Select one or more objectives for your design and explain how you plan to further optimize it for each.

Justification

One objective that I want to solve is the weight issue, how I could further optimize it is by making the width of the objects smaller because as it is right now the width is way larger than it needs to be. By reducing the width it uses less material, adds less friction and also makes it weigh less.

MILESTONE 3 (INDIVIDUAL) – COVER PAGE

Please list full name and MacID.

Full Name:	**MacID:**
Shuja Wazir	wazirs4

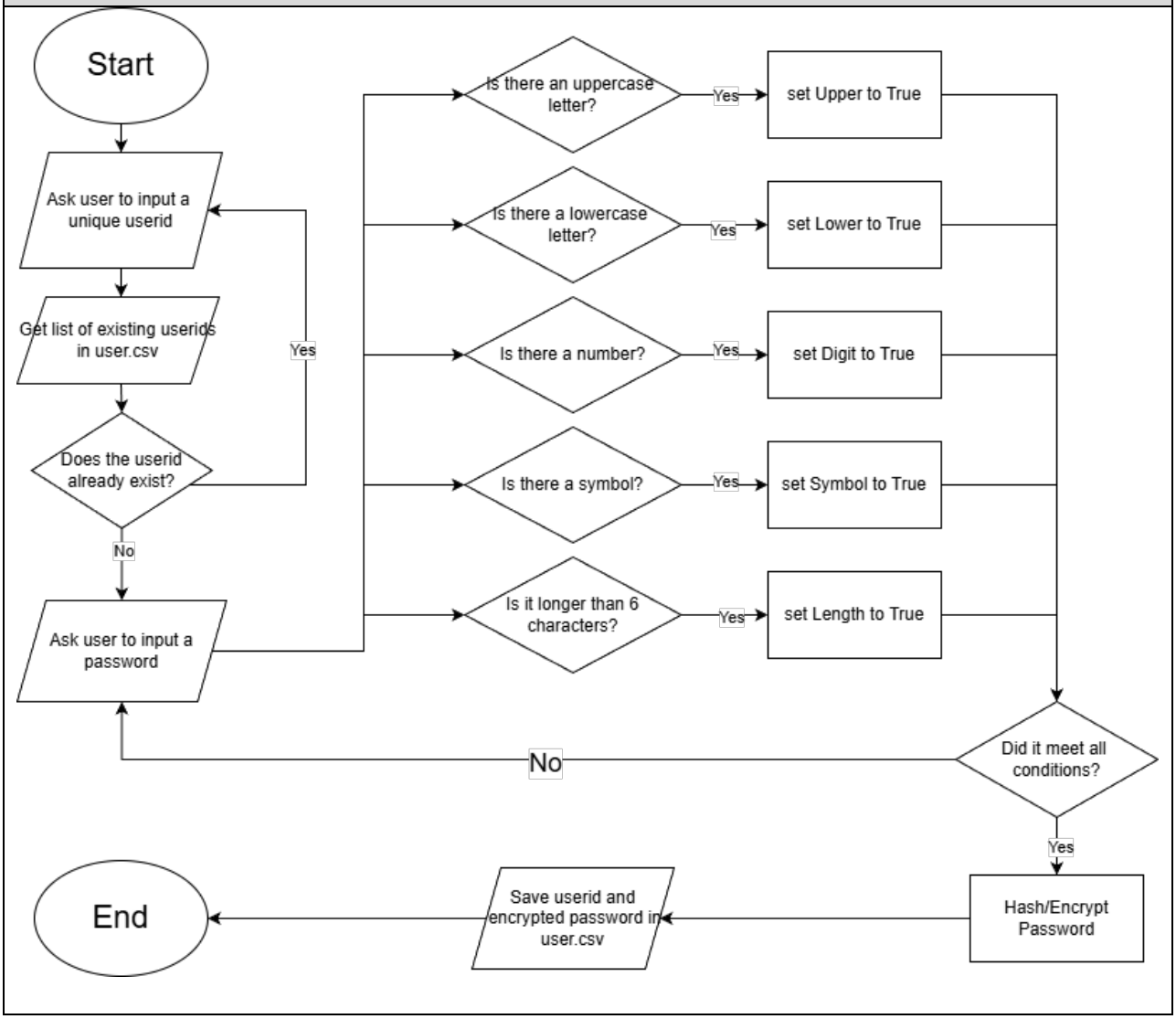
MILESTONE 3 (STAGE 2) – FLOWCHART

Team ID:

Tues-52

Insert an image of flowchart in the box for your assigned function.
Also change the name of the function to match your assigned function.

sign_up()



Milestone 1 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

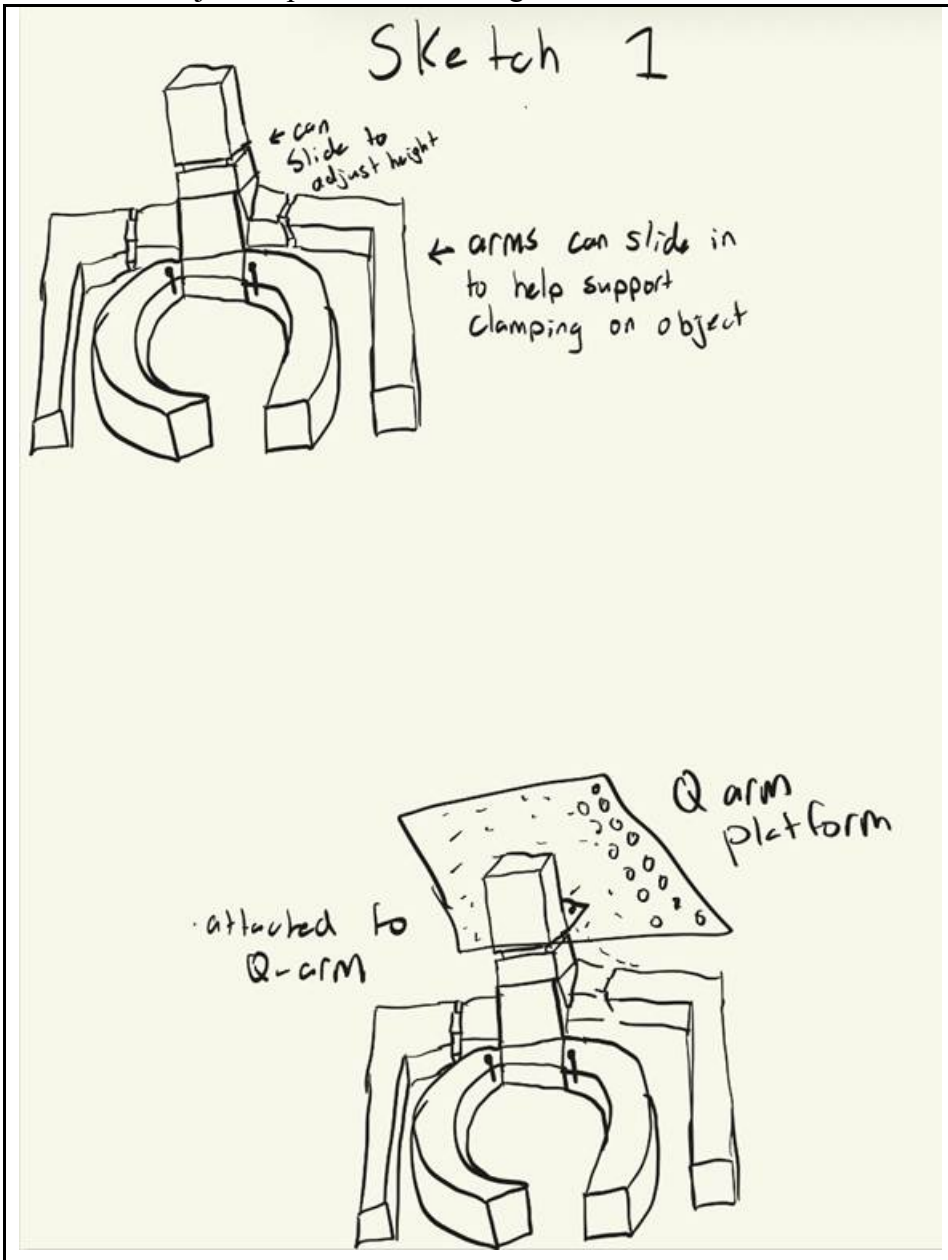
Full Name:	MacID:
Jennifer	Desbarats

Milestone 1 (Stage 2) – Mechanism Concept sketches

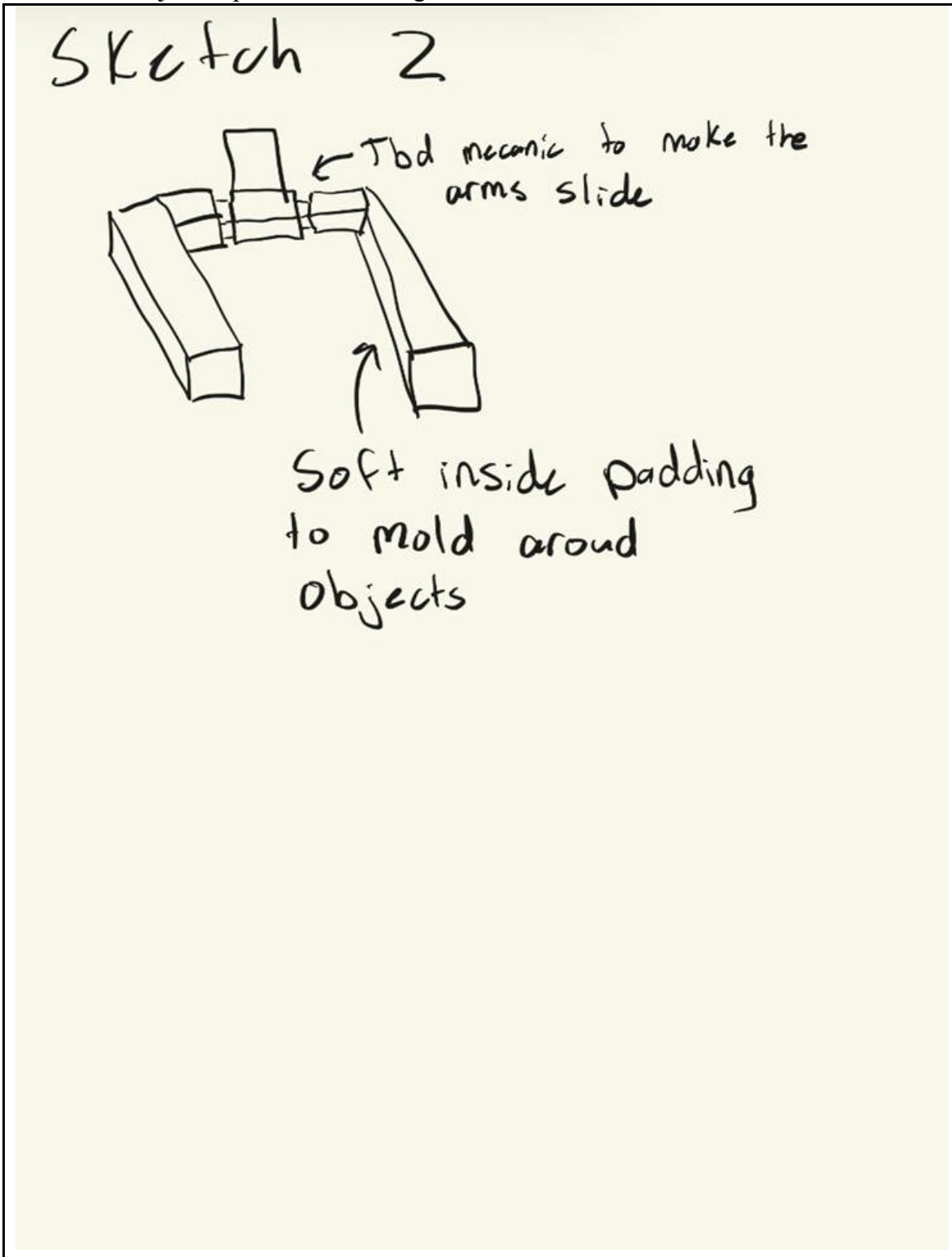
Team ID: Day-##

Insert images of your initial mechanism sketches in the box below.

Sketch #1


Sketch 1
← can slide to adjust height
← arms can slide in to help support clamping on object
Q arm platform
· attached to Q-arm

Sketch #2



Select one or more objectives for your design and explain how you plan to optimize it for each.

Justification

Objective for design is to be able to effectively pick up objects of different sizes, shapes and rigidity. In my second design, I put foam or a soft material on the inside of the grips to be able to properly mold around each object with a more malleable material.

Another objective for design is to be able to securily pick up objects. My first design has arms that are circular to mold around objects. To properly secure the object, a sliding pair of straight arms will come it to solidify the hold and potentially apply more support.

Individual Worksheet (Jen):

Milestone 2 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

Full Name:	**MacID:**
Jennifer Desbarats	desbaraj

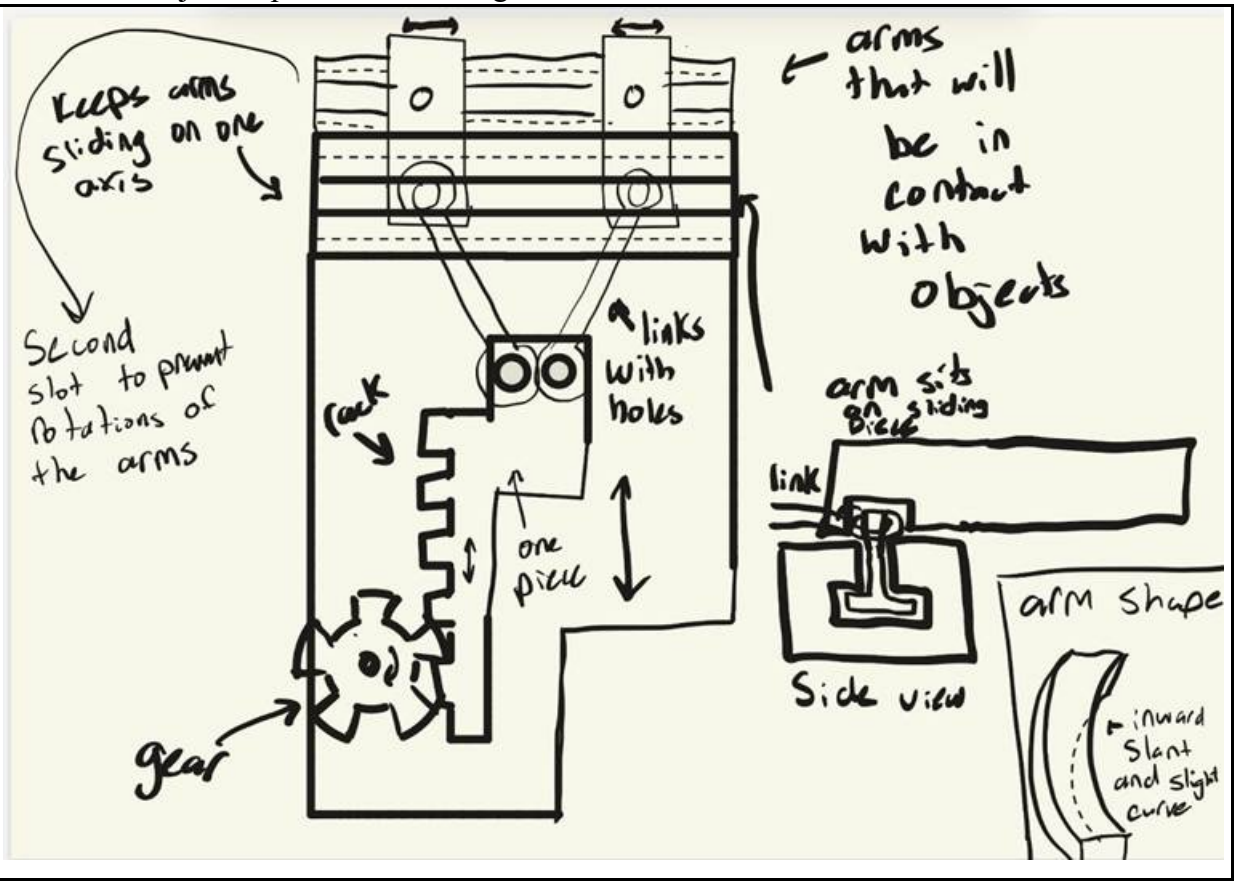
Milestone 2 (Stage 1) – Finalized Concept Sketches and CAD Model worksheet

Team ID: Day-##

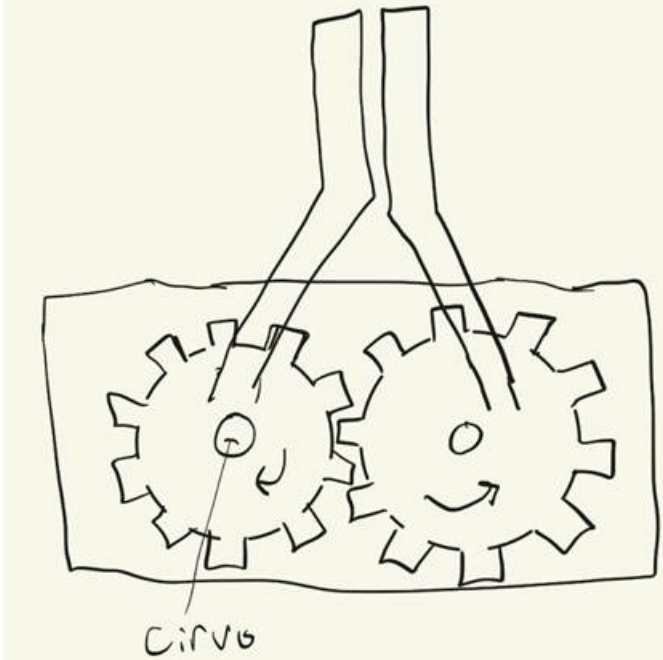
This is an individual deliverable and should be completed by each team member **prior** to Design Studio.

Attach images to the following three boxes of your mechanism sketch concepts and ideas.

Refined Sketch #1

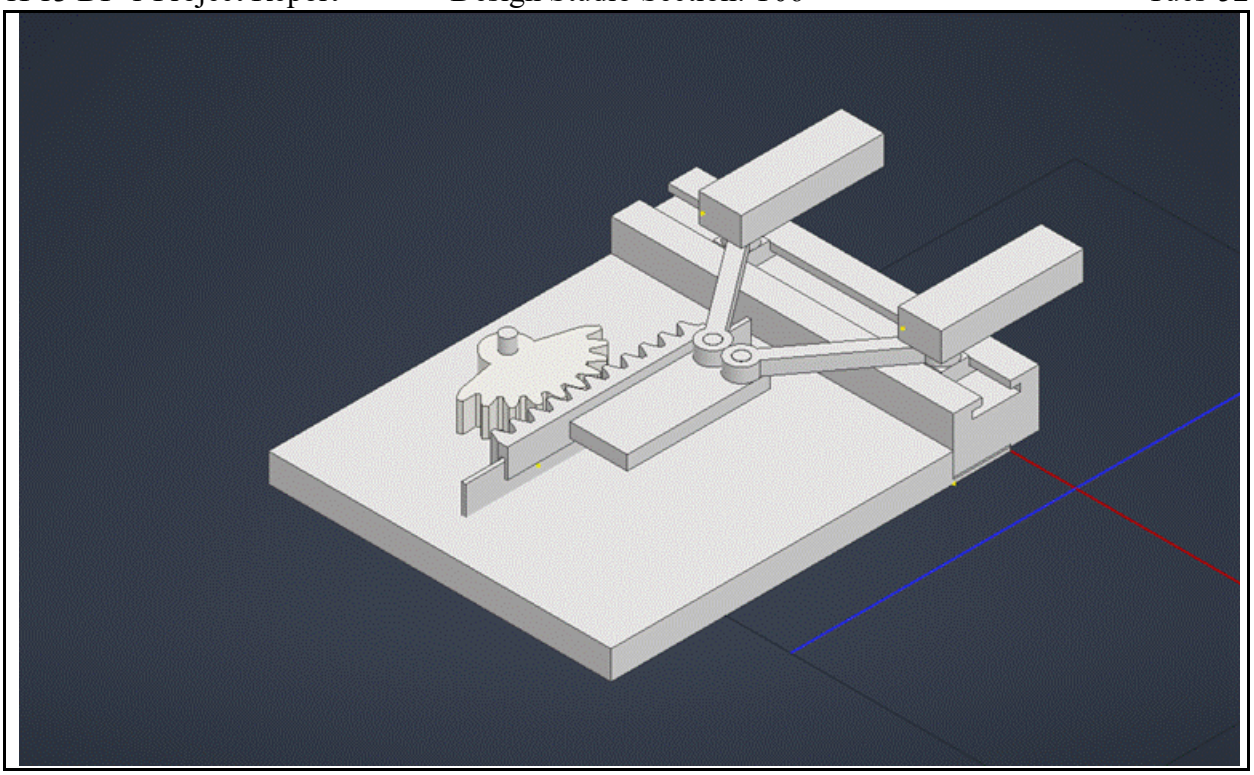


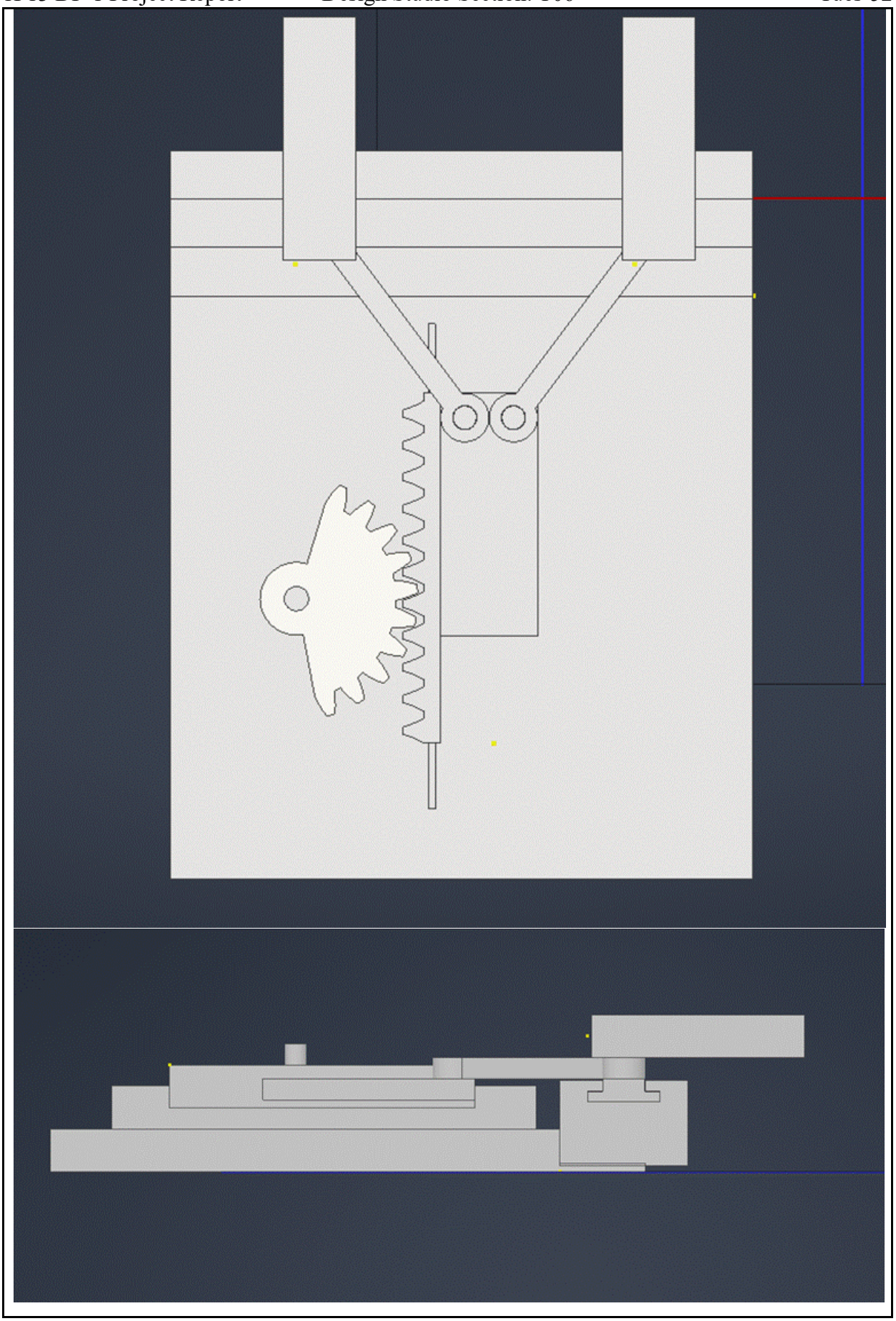
Refined Sketch #2



CAD Model Screenshots

Include screenshots of the top, front, right and isometric views of your CAD model. All design features of your CAD model should be visible.





Select one or more objectives for your design and explain how you plan to further optimize it for each.

Justification
An issue with my design is that the arms will rotate when they come into contact with an object. To fix this, we need to add a second contact point as with 2 contact points it wont rotate. A second slot could be added with a second contact point to fix this. Another objective with my design is to be light. To further optimize this, material from the base could be cut out to make the whole design lighter.

Milestone 3 (Individual) – Cover Page

Team ID: Tues-52

Please list full name and MacID.

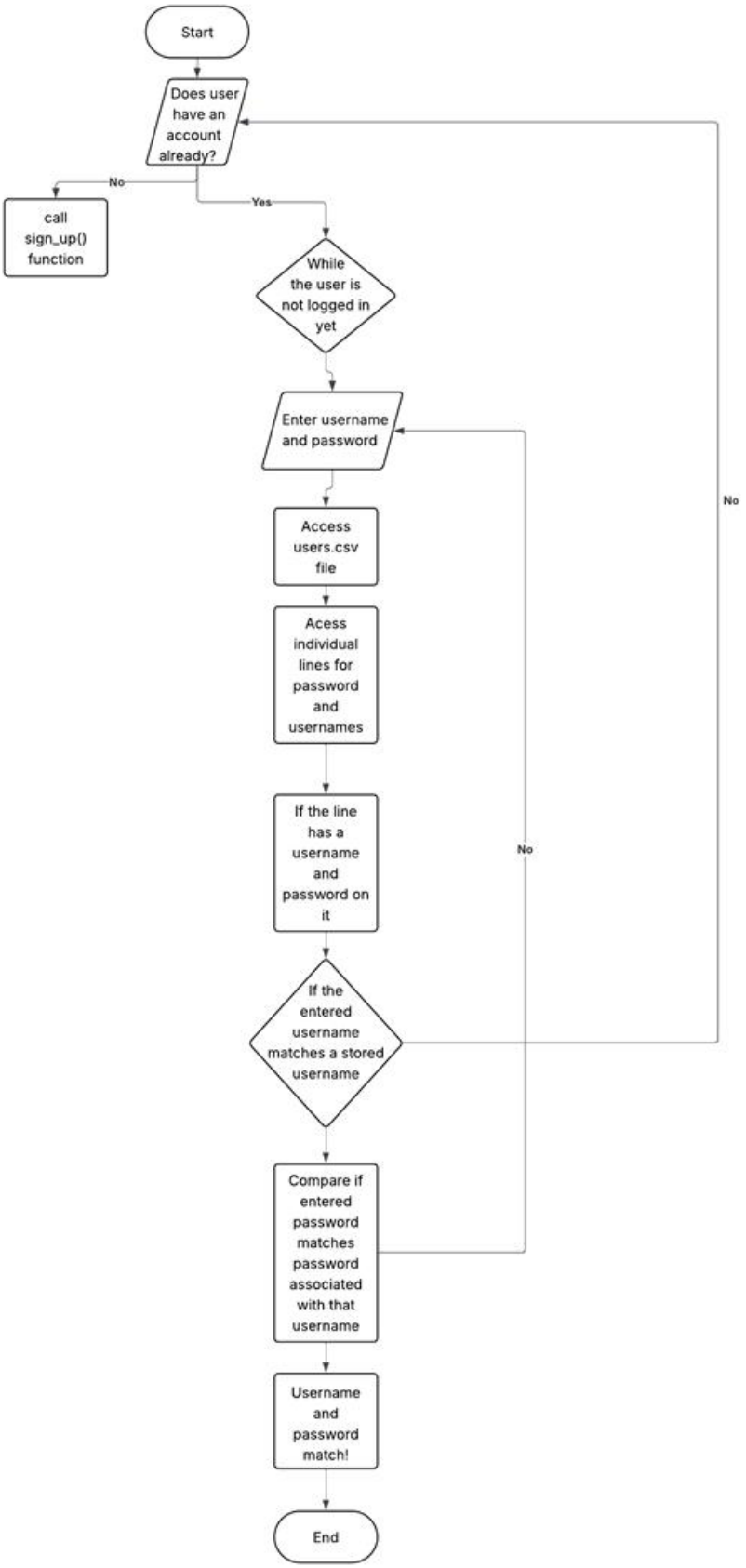
Full Name:	**MacID:**
Jennifer Desbarats	desbaraj

Milestone 3 (Stage 2) – Flowchart

Team ID: Day-##

Insert an image of flowchart in the box for your assigned function. Also change the name of the function to match your assigned function.

authenticate()


Start
Does user have an account already?
No
Yes
call sign_up() function
While the user is not logged in yet
Enter username and password
Access users.csv file
Acess individual lines for password and usernames
If the line has a username and password on it
If the entered username matches a stored username
Compare if entered password matches password associated with that username
Username and password match!
End
No
No

MILESTONE 1 (INDIVIDUAL) – COVER PAGE

Please list full name and MacID.

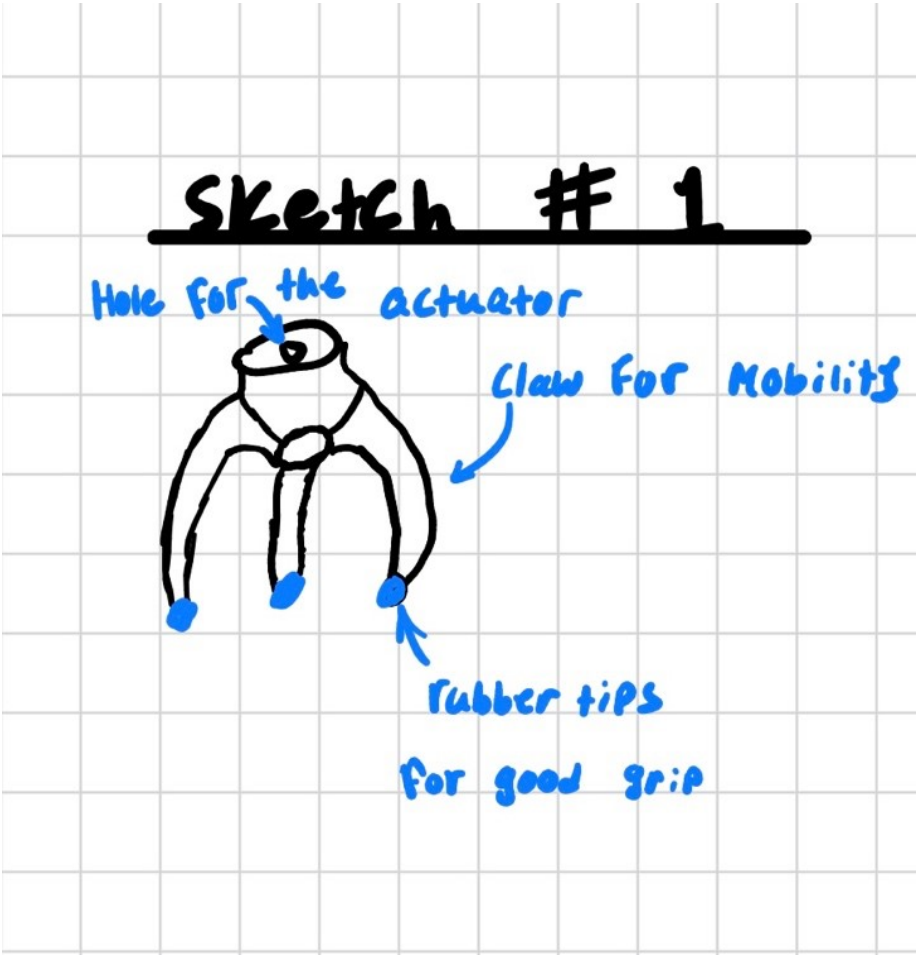
Full Name:	**MacID:**
Shayan Siddiqui	Siddm47

MILESTONE 1 (STAGE 2) – MECHANISM CONCEPT SKETCHES

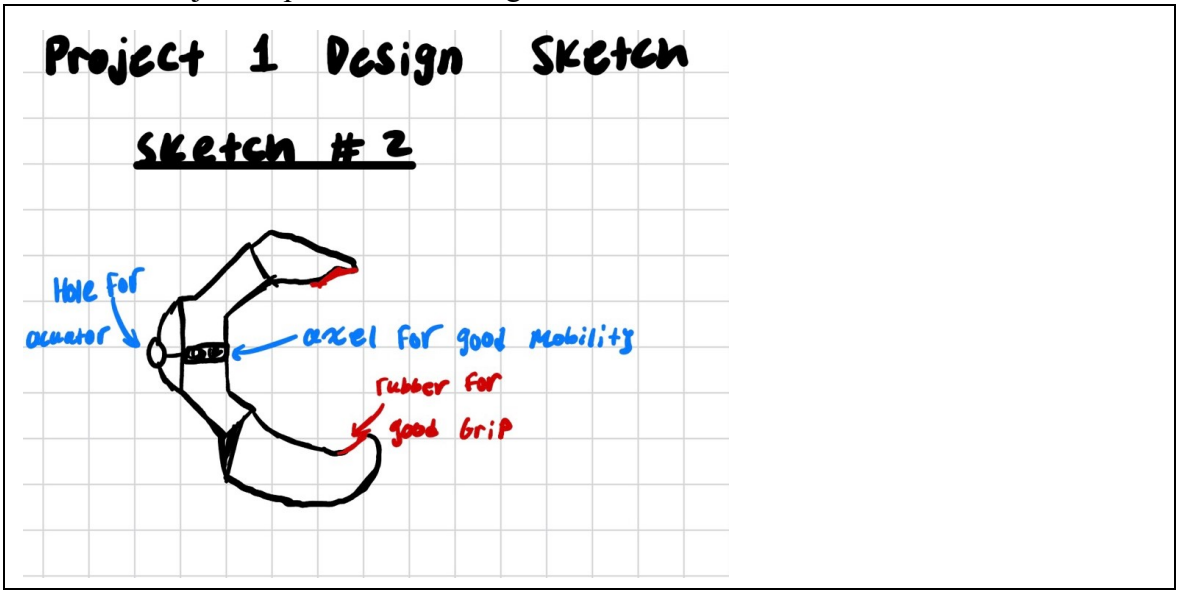
Team ID: Tues -52

Insert images of your initial mechanism sketches in the box below.

Sketch #1



Sketch #2



Select one or more objectives for your design and explain how you plan to optimize it for each.

Justification
One of the objectives are that it should have a good grip. This is achieved by the rubber ends on the claws to make sure that the designated items don't slip out of the claw. Another objective was to have it be versa6le so that it can pick up unique items. This is accounted for in sketch 1 since having 3 fingers on the claw which allows for the claw to be able to pick up different kinds of objects.

MILESTONE 2 (INDIVIDUAL) – COVER PAGE

Tues-52

Please list full name and MacID.

Full Name:	MacID:
Shayan Siddiqui	Siddm47

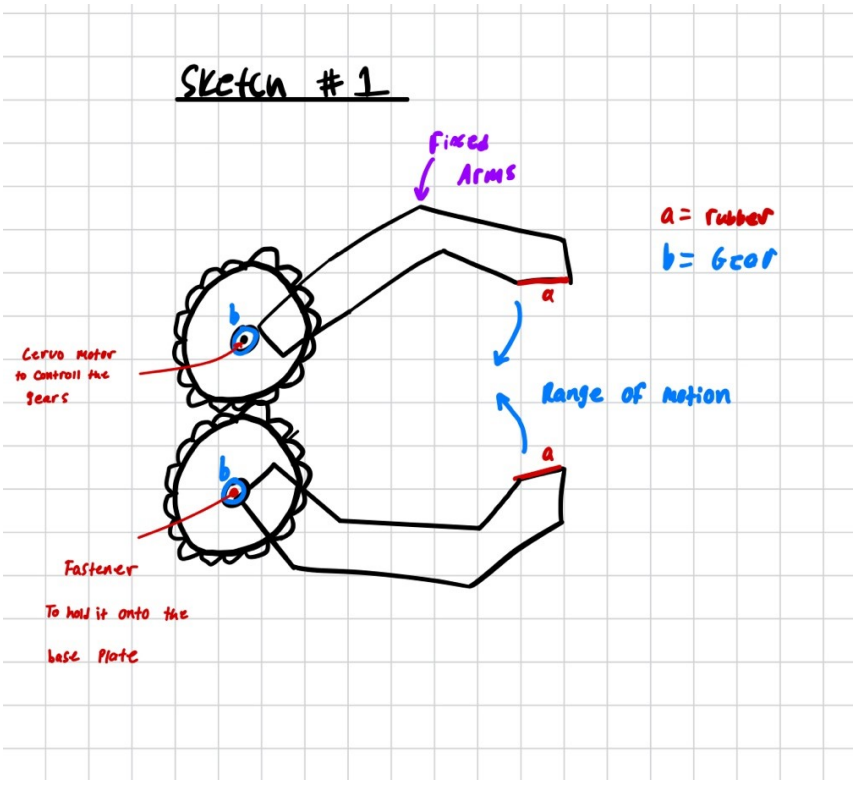
MILESTONE 2 (STAGE 1) – FINALIZED CONCEPT SKETCHES AND CAD MODEL WORKSHEET

Team ID: Tues-52

This is an individual deliverable and should be completed by each team member **prior** to Design Studio

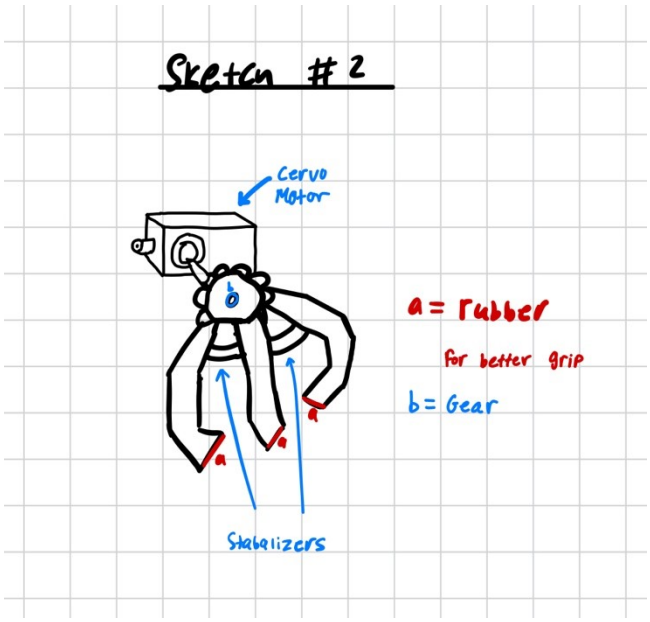
Attach images to the following three boxes of your mechanism sketch concepts and ideas.

Refined Sketch #1



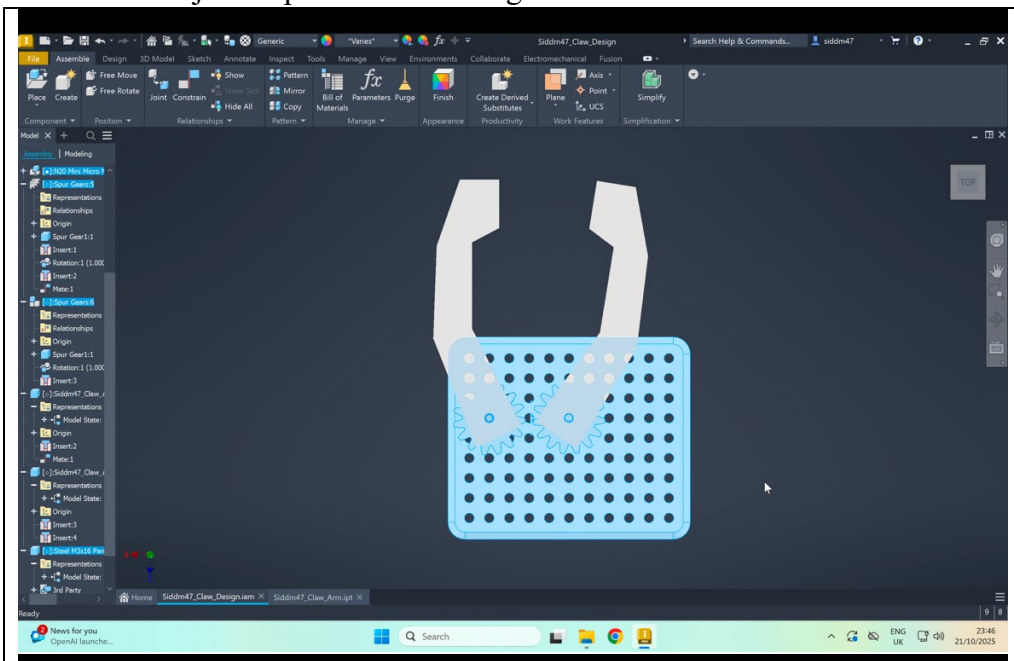
Tues-52

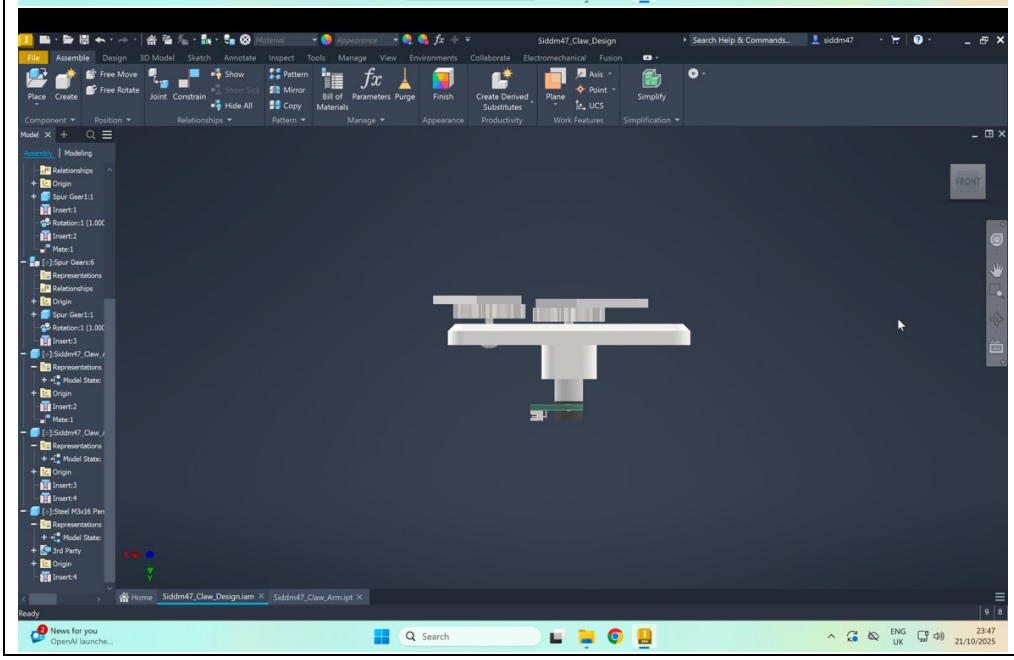
Refined Sketch #2



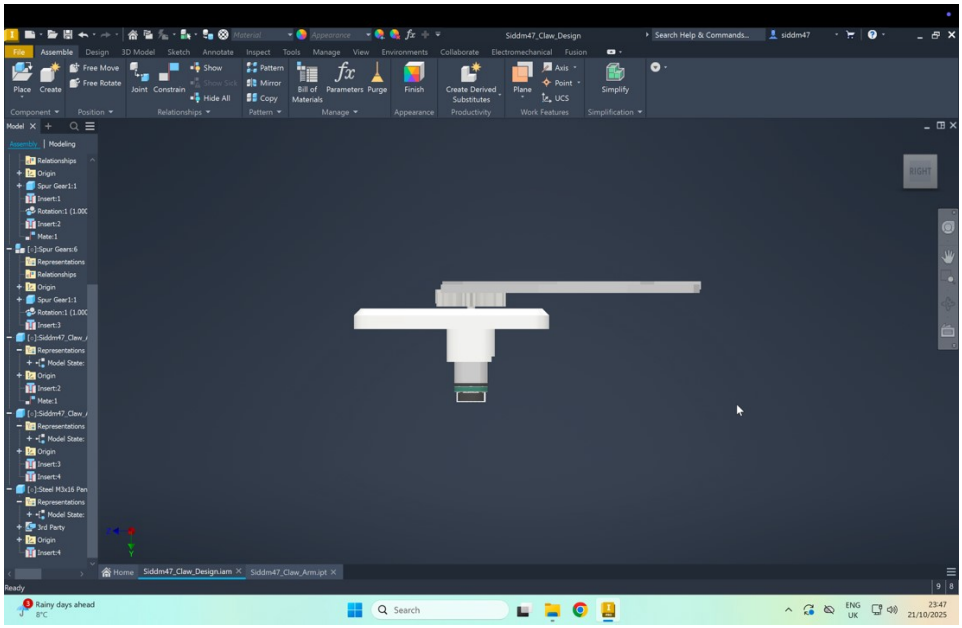
CAD Model Screenshots

Include screenshots of the top, front, right and isometric views of your CAD model. All design features of your CAD model should be visible.





Tues-52



Select one or more objectives for your design and explain how you plan to further optimize it for each.

Justification
An objec4ve for my design is that it should have a strong grip. After my TA check in I got some feedback that the hands of my claws should maybe be a bit thicker so that it could be more sturdy and handle heavier objects and be able to hold on to the items without breaking. Another objective is that it should be able to pick up a variety of different objects. Again this could be optimized by perhaps adding some angles to the arms of the claw so it could get under the objects and be able to pick up more items.

MILESTONE 3 (INDIVIDUAL) – COVER PAGE

Team ID:

Please list full name and MacID.

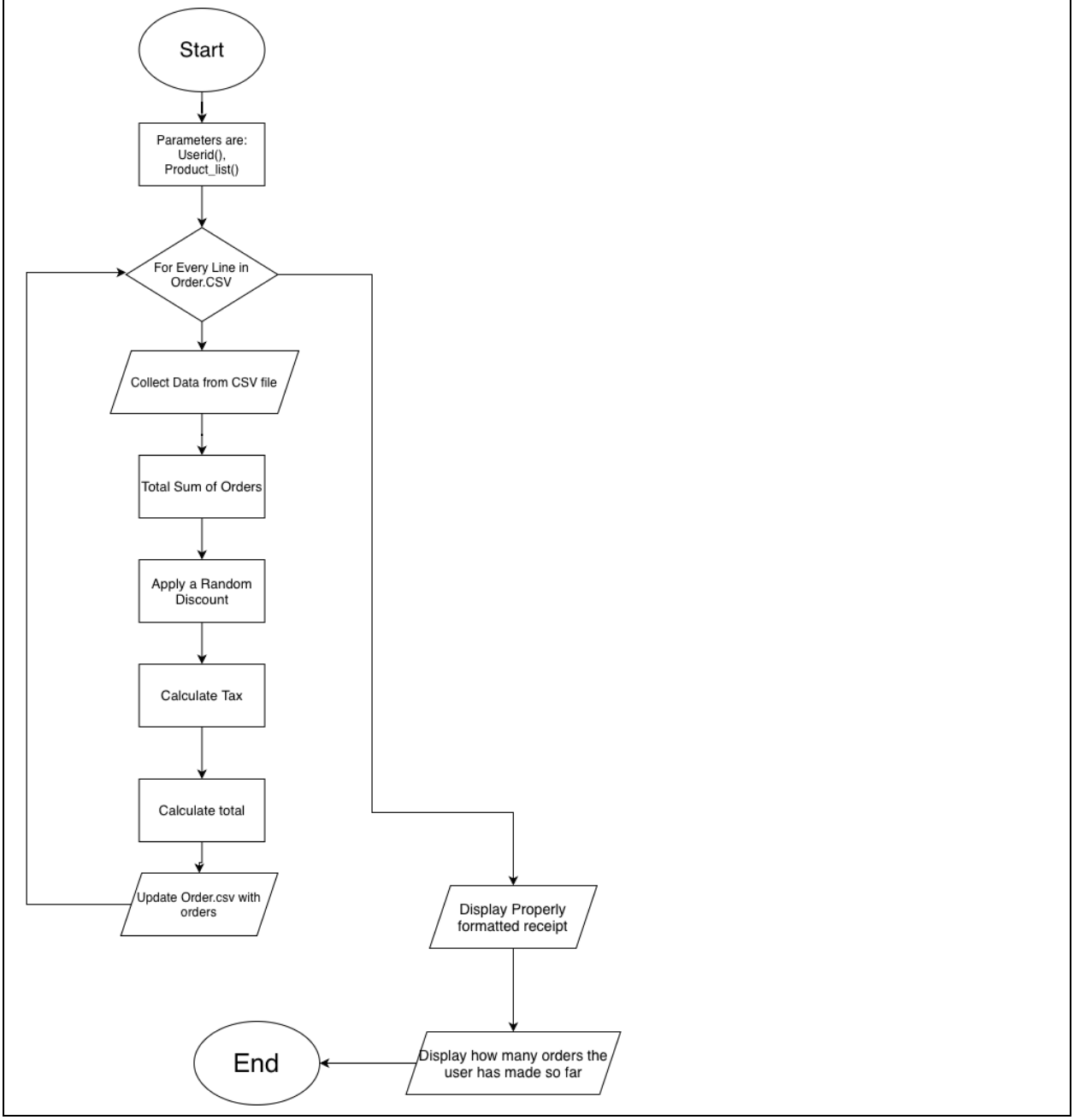
Full Name:	MacID:
Shayan Siddiqui	Siddm47

MILESTONE 3 (STAGE 2) – FLOWCHART

Team ID: Tues-52

Insert an image of flowchart in the box for your assigned function. Also change the name of the function to match your assigned function.

complete_order()



Milestone 1 (Individual) – Cover Page

Team ID: Tues-52

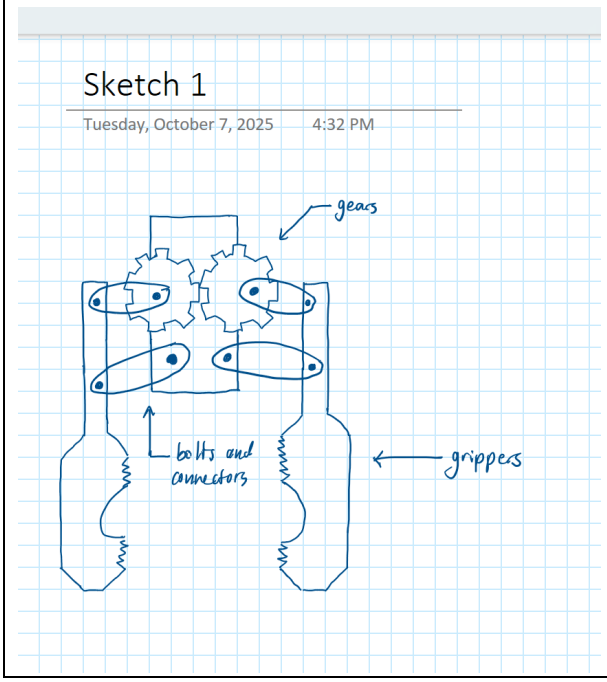
Please list full name and MacID.

Full Name:	MacID:
Habibah Aboueleinin	Aboueh2

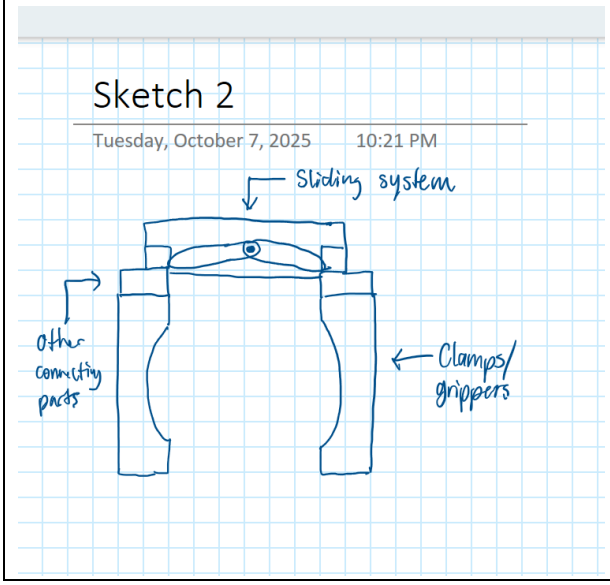
Team ID: Tues-52

Insert images of your initial mechanism sketches in the box below.

Sketch #1



Sketch #2



Select one or more objectives for your design and explain how you plan to optimize it for each.

Justification
One of the objectives my group and I discussed was that it should be versatile to pick up different shapes. In my sketches, I specifically chose to add those inner rounded parts to the clamps/grippers to expand the options of what this system can retrieve. In sketch one, having a rigid edge on the grippers also helps minimize friction making it easier to pick up objects with smoother surfaces. In sketch two, the sliding feature and the round edge make a perfect team to carry small objects and different shapes.

Team ID: Tues-52

Please list full name and MacID.

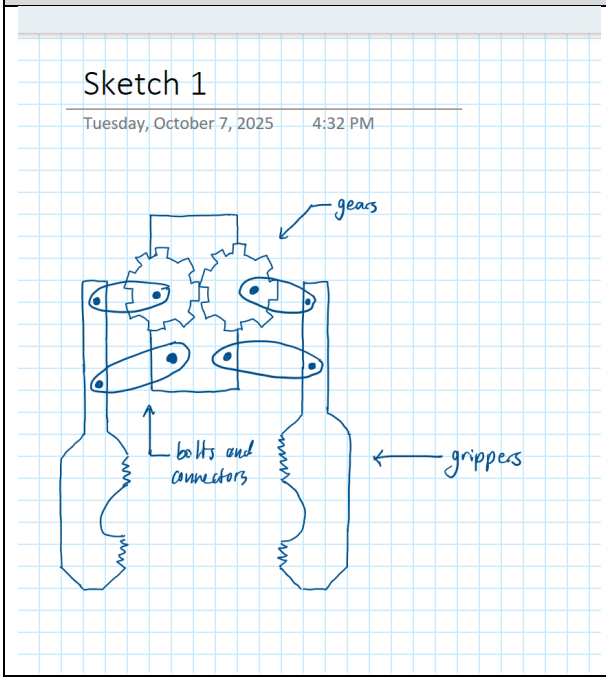
Full Name:	MacID:
Habibah Aboueleinin	Aboueh2

Team ID: Day-##

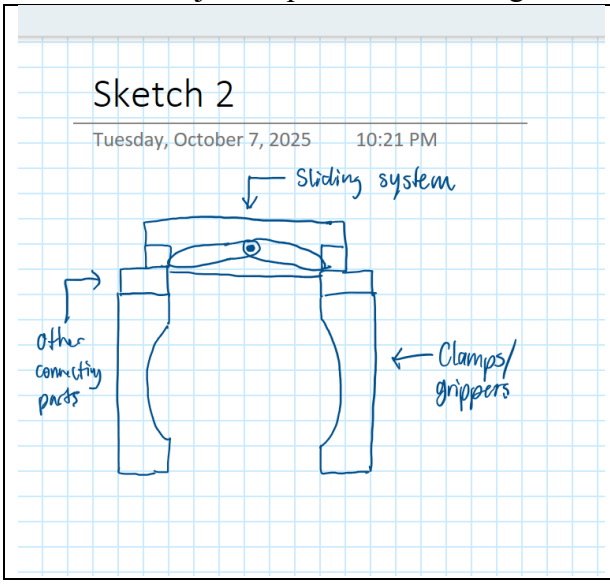
This is an individual deliverable and should be completed by each team member **prior** to Design Studio.

Attach images to the following three boxes of your mechanism sketch concepts and ideas.

Refined Sketch #1

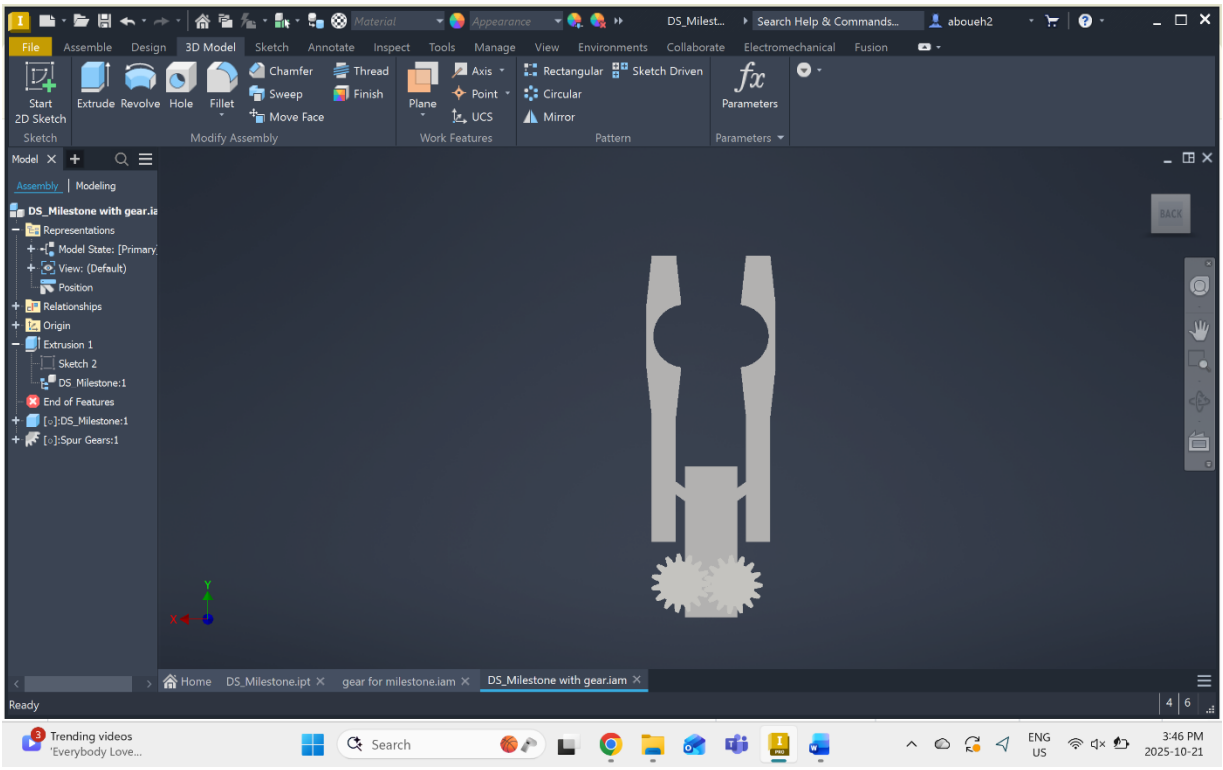


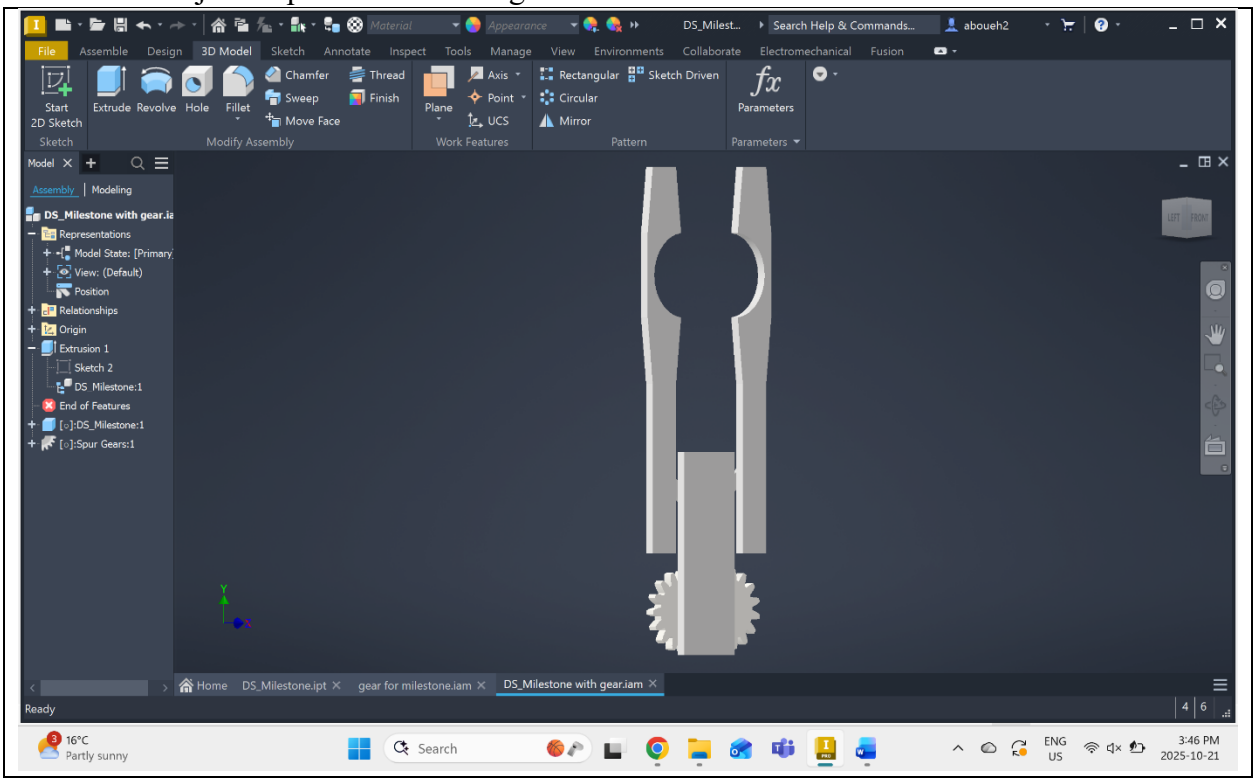
Refined Sketch #2



CAD Model Screenshots

Include screenshots of the top, front, right and isometric views of your CAD model. All design features of your CAD model should be visible.





Select one or more objectives for your design and explain how you plan to further optimize it for each.

Justification
One of the objectives my group and I discussed was that it should be versatile to pick up different shapes. In my sketches, I specifically chose to add those inner rounded parts to the clamps/grippers to expand the options of what this system can retrieve. In sketch one, having a rigid edge on the grippers also helps minimize friction making it easier to pick up objects with smoother surfaces. In sketch two, the sliding feature and the round edge make a perfect team to carry small objects and different shapes.

Team ID: Tues-52

Please list full name and MacID.

Full Name:	MacID:
Habibah Aboueleinin	Aboueh2

Team ID: Tues-52

Insert an image of flowchart in the box for your assigned function. Also change the name of the function to match your assigned function.

customer_summary()