B2: Smart Wardrobe

Problem Area:

- Picking the right outfit can be a difficult task
  - Difficult and time consuming to test outfit combinations manually
  - Hard to visualize how new articles of clothing will add to a wardrobe
  - Outfits need to account for day to day situations
- As wardrobes get bigger, they become more difficult to manage
  - Difficult and time consuming to find articles manually

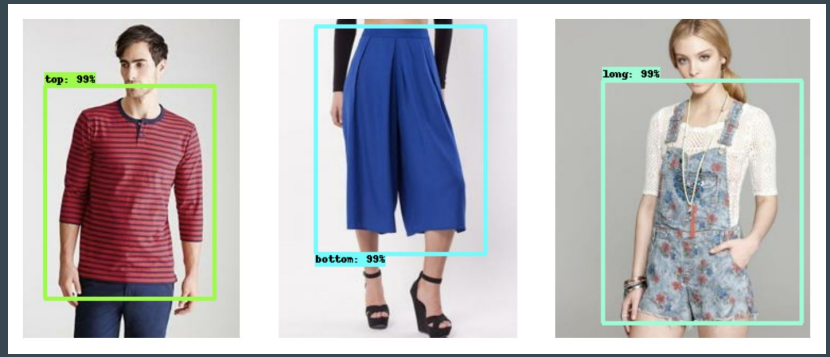
Henry Lin, Sung Hyun Back, Fred Yoojoon Lee

Use case

- Outfit Visualizer (software)
  - Recognizes client's clothing articles and finds matching/similar outfits online
  - Client can pick specific clothing articles (pants, shirt, or jacket) for the visualizer to search on
  - Other basic filters like color, type, temperature, etc.
- iButler (hardware)
  - Can automatically retrieve clothing articles specified via simple user interface
  - Can automatically store clothing articles

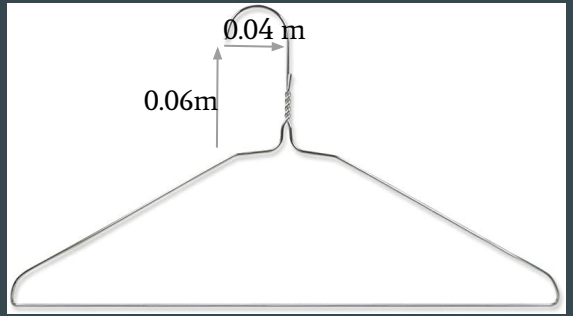
Software Requirements

- Clothing Matching
  - S1. Can find x amount of trending outfit photos that match user-given specifications
  - S2. Continuously adds new outfit photos as new ones are posted online
- Clothing Recognition
  - S3. Using the photos of outfits it can determine pants, shirts, and jackets
  - S4. Identifies color and variation of clothing, like jeans or shorts
  - S5. Maintain a minimum of x% accuracy in identifying clothes
- Runtime
  - S6. Software should be able to execute operations within a few seconds



Hardware Requirements

- Horizontal Movement
  - H1. Move to the designated coordinates within ± 0.02m
- Vertical Movement (grabbing and placing the clothes)
  - H2. Vertical Margin of error within ± 0.03m
- Retrieval Speed
  - H3. Retrieve/store clothing articles within 10 seconds
- Clothing Requirements
  - C4. Clothing articles are placed 10 cm apart
  - C5. Clothes weigh less than 2 kg



Technical Challenges (Software)

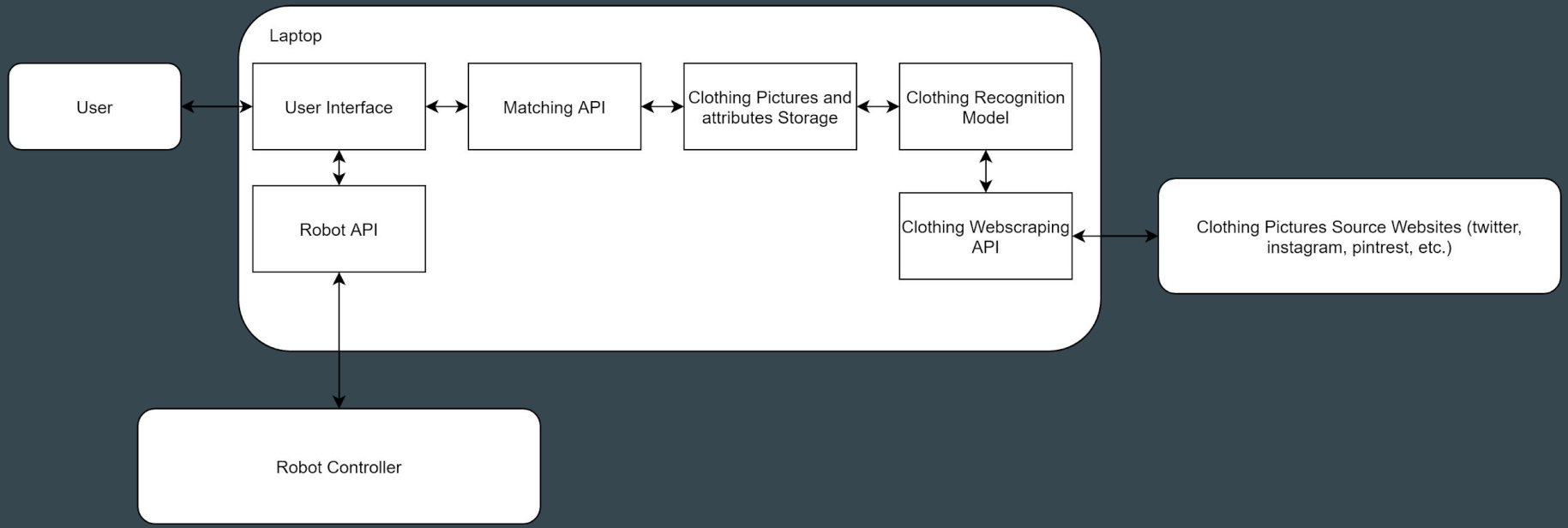
- Clothing Matching
  - Runtime: Could be difficult to find the amount requirement within runtime requirement if outfit is difficult to find online
    - RM: mitigate latency with good caching and precomputing
- Clothing Recognition
  - Model must correctly label outfits with high precision
  - Many varying attributes like lighting, pose, and occlusion could be a challenge for the model
    - RM: use a predefined dataset with outliers removed if necessary

RM == Risk Mitigation

Technical Challenges (Hardware)

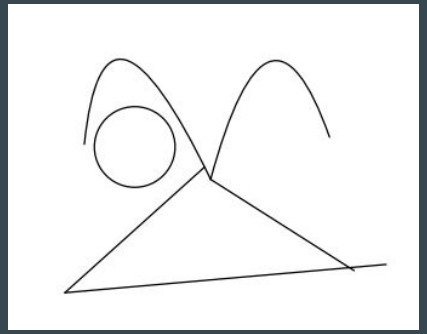
- Accuracy
  - The clothing hanger hook has a small surface area so we have to be accurate
    - RM: good hardware design with minimal margins of error
  - Have to ensure the right clothing article is chosen
    - RM: Correct calibration and error correction in software
- Weight
  - Robot arm should be able to handle weight up to 2kg
    - RM: sturdy motors/actuators that won't malfunction when its carrying clothes
  - Consistency in movement speed and precision regardless of having clothes or not
    - RM: Not too sure what this will be

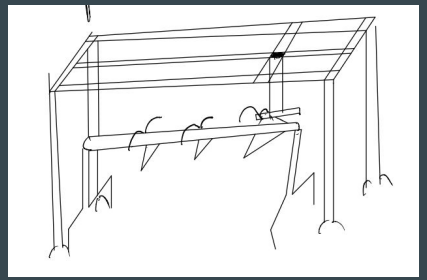
Solution Approach (Software)

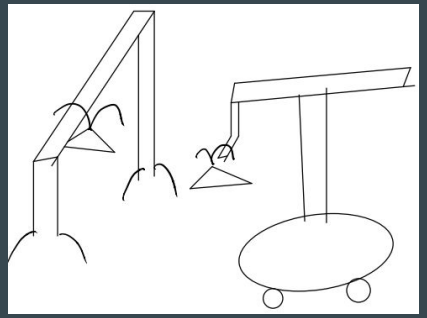


Solution Approach (Hardware)

- Fixed rail device (Claw Machine)
  - Sliding claw machine with 3 degrees of motion (x, y, z)
  - Smaller margin of error compared to modular approach
  - Requires building rail system that could be difficult to implement
- Modular device (Roomba)
  - Roomba provides 2 degrees of motion (x,y)
  - Third degree of motion from linear actuator
  - Easier to implement in terms of hardware
  - Requires error correction software
  - Higher margin of error







Testing, Verification, and Metrics (Software)

- Clothing Matching
  - S1. Automated test cases that uses exercises matching API to reach targeted amount of outfit photos
  - S2. Ensure photos are from online source (attach URL)
- Clothing Recognition
  - S3/4/5. DeepFashion dataset contains over 800,000 labeled fashion images that we can use for both training and testing
    - Because it already has the labels we want, we can use it to run validation testing
- Runtime
  - Benchmark tests for varying predefined and random test cases

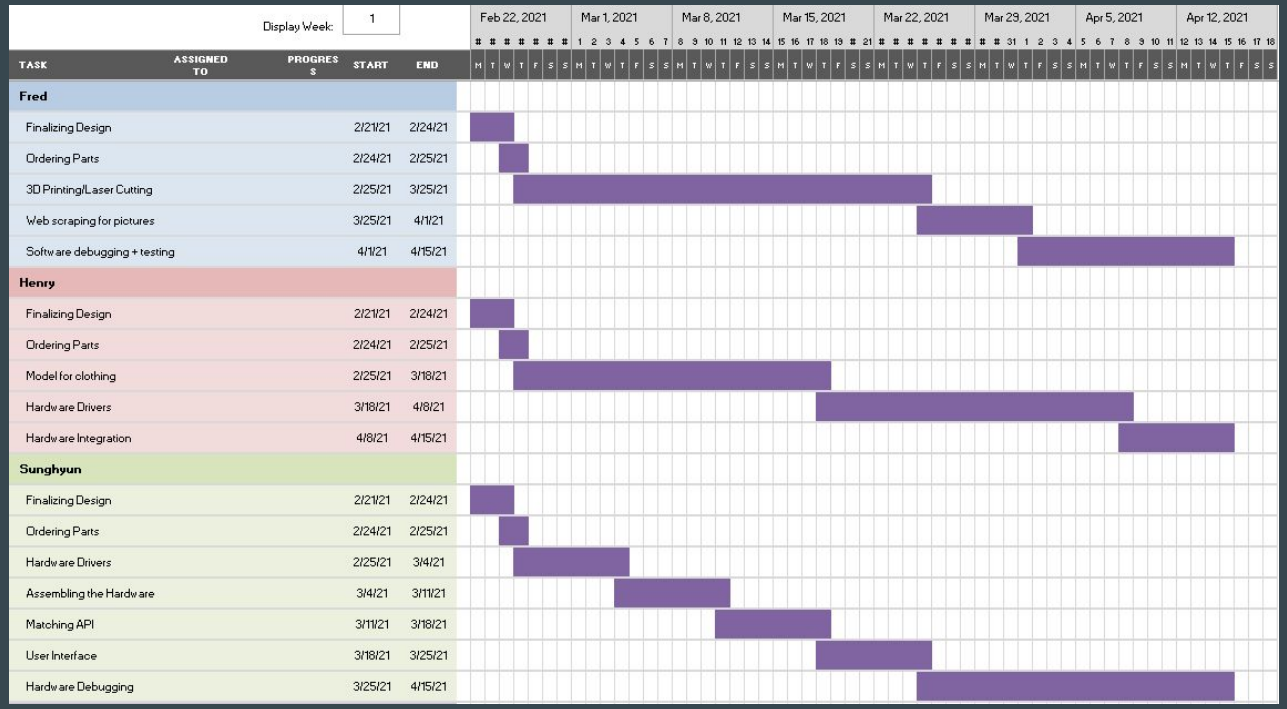
Testing, Verification, and Metrics (Hardware)

- Horizontal Movement
  - Can move on the x and y axis consistently to .02 m of the location
  - If we can maintain this accuracy in 19 out of 20 trials
  - Test specifications:
    - Designated locations are randomly selected out of a fixed grid system
    - Tape defining the margin of error can be used to quickly identify if a test passes or not
- Vertical Movement
  - Can move on the vertical axis to within .03 m of the location
  - If we can maintain this accuracy in 19 out of 20 trials
  - Similar test specifications to horizontal movement
- Load Testing
  - Adding load to our horizontal and vertical movement tests
  - Testing with 0 kg load, 1 kg load, and 2 kg load

Tasks and Division of Labor

- Hardware
  - HT1. Finalize design of our hardware + controller (3 days, Team)
  - HT2. Order necessary parts online (1 day, Team)
  - HT3. 3D Printing / laser cutting (depending on our method) (3 weeks, Fred)
  - HT4. Assembling the hardware (1 week, Sung Hyun)
  - HT5. Hardware drivers (4 weeks, Henry, Sung Hyun)
  - HT6. Hardware debugging and testing (3 weeks, Sung Hyun)
- Software
  - ST1. Web scraping for pictures (1 week, Fred)
  - ST2. Model to identify clothes (3 weeks, Henry)
  - ST3. Hardware integration (1 week, Henry)
  - ST4. Matching API (1 week, Sung Hyun)
  - ST5. User interface (1 week, Sung Hyun)
  - ST6. Software debugging and testing (2 weeks, Fred)

Schedule



Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Fred				
Finalizing Design			2/21/21	2/24/21
Ordering Parts			2/24/21	2/25/21
3D Printing/Laser Cutting			2/25/21	3/25/21
Web scraping for pictures			3/25/21	4/1/21
Software debugging + testing			4/1/21	4/15/21
Henry				
Finalizing Design			2/21/21	2/24/21
Ordering Parts			2/24/21	2/25/21
Model for clothing			2/25/21	3/18/21
Hardware Drivers			3/18/21	4/8/21
Hardware Integration			4/8/21	4/15/21
Sunghyun				
Finalizing Design			2/21/21	2/24/21
Ordering Parts			2/24/21	2/25/21
Hardware Drivers			2/25/21	3/4/21
Assembling the Hardware			3/4/21	3/11/21
Matching API			3/11/21	3/18/21
User Interface			3/18/21	3/25/21
Hardware Debugging			3/25/21	4/15/21

Feb 22, 2021 | Mar 1, 2021 | Mar 8, 2021 | Mar 15, 2021 | Mar 22, 2021 | Mar 29, 2021 | Apr 5, 2021 | Apr 12, 2021