UMassAmherst
The Commonwealth's Flagship Campus

Preliminary Design Review: ASOS

Automated Screw Organization System

Team 20
Jordon Balskus, Jordan Gyaltsen, Andrew McGrath, Rajesh Shahi

October 15, 2018

Advisor: Professor Eslami

Outline

1. Problem Statement
2. System Specifications
3. Design System
4. Team Roles
5. MDR Deliverables

Problem Statement

- There are ~17,000 machines shops across the United states that will inevitably have to deal with one problem, time after time. Unsorted screws.

- Current solution to dealing with a bucket of random screws
  - Throw them out or manually spend hours checking the diameter, thread spacing, and, length for possibly several different types of screws by measurement or individual threading holes.
  - Throwing out screws contributes to growing trash problem that our planet faces and is economically wasteful

Design Alternatives (Existing Products)

- **LS52-100 - SCREW & CONNECTOR BAG**
  - Heavy and inconvenient to carry around
  - Have to remember which screw is in which pocket
  - No guarantee of sorted screw

- **PS 1500**
  - Very expensive (~ $1,013 - $ 2,000)
  - Not portable

Design Alternatives (Existing Products)

- **Ray's Screw Sorting Machine**
  - Inefficient, user has to put the screw one by one
  - Limited options
  - Very noisy, unpleasant to the user

Characteristics Of A Screw

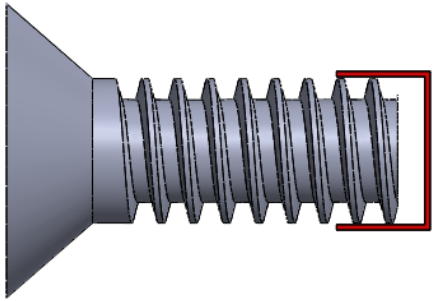
Imperial screw Labeling : # - TC x L

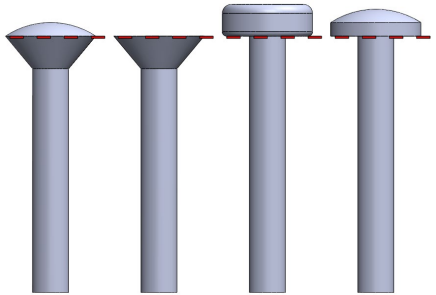
Diameter

- Distance through the edges of one thread
- $D = (\# * 0.013) + 0.06$

Length

- L = Distance from bottom of head to bottom of screw

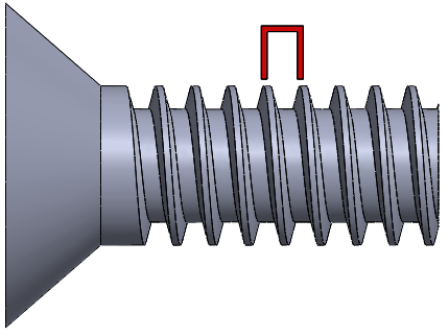




Characteristics Of A Screw

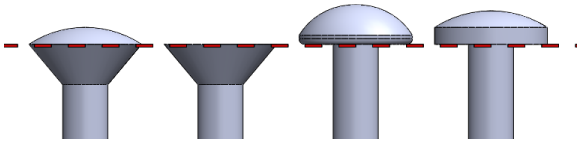
Thread Count

- Number of threads per square inch



Head Type

- Same screws with different heads



Proposed Solution

- Three tiered system to separate, identify, & sort the different types of commonly used screws
- Utilize distinguishable characteristics of screws to identify and sort limited number of screws
- Compact & affordable design compatible for all small & mid-level machine shops

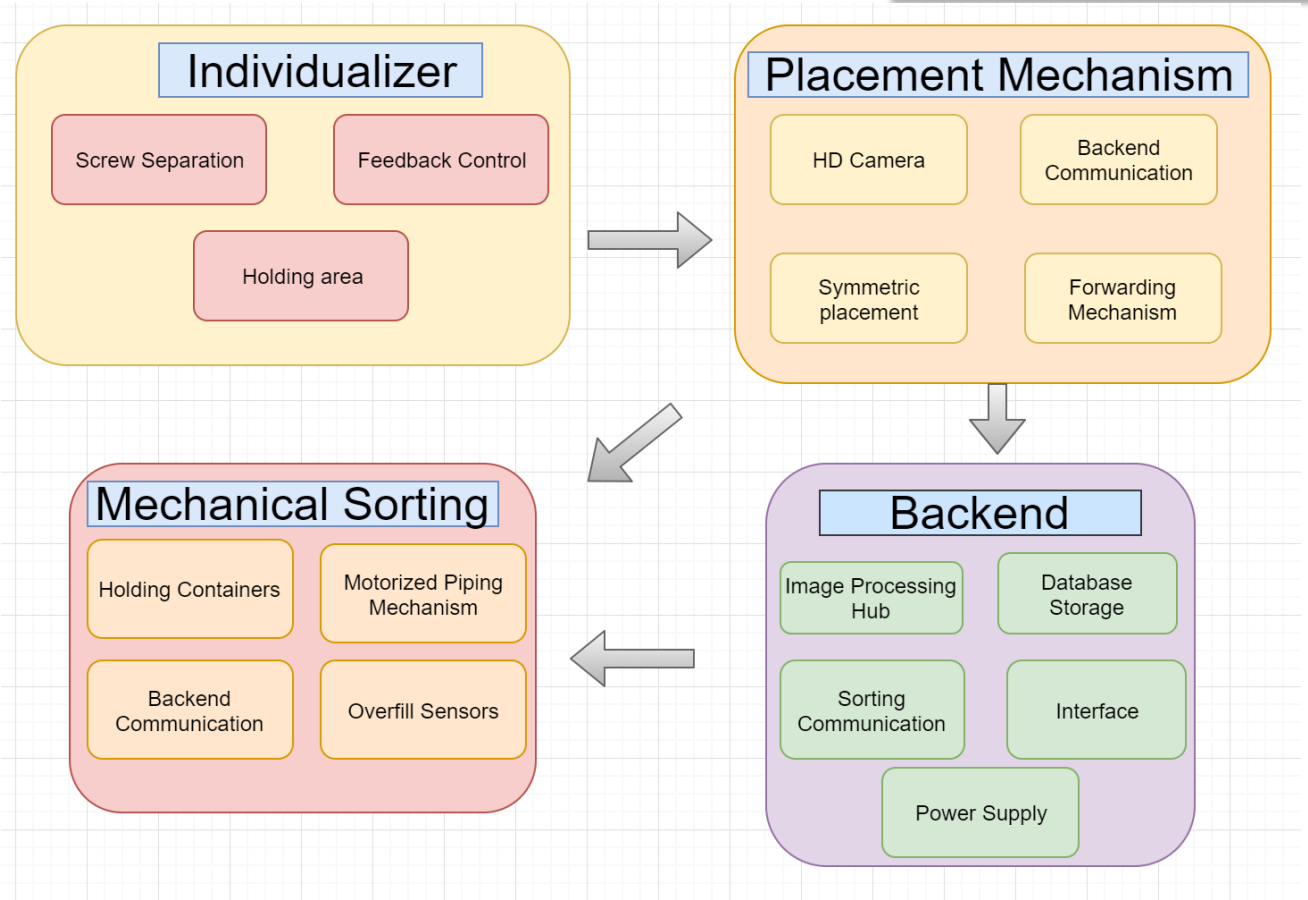
Requirement Analysis: Specifications

- Screw length between 6mm & 100mm
  - limit scope of mechanical complications
  - works for most screws in machine shop
- Sort between 5-10 different types of screws at a time
  - Excess screws placed together then resorted
- Real-Time interface communication
  - Screws easily identified once sorted

Requirement Analysis: Specifications

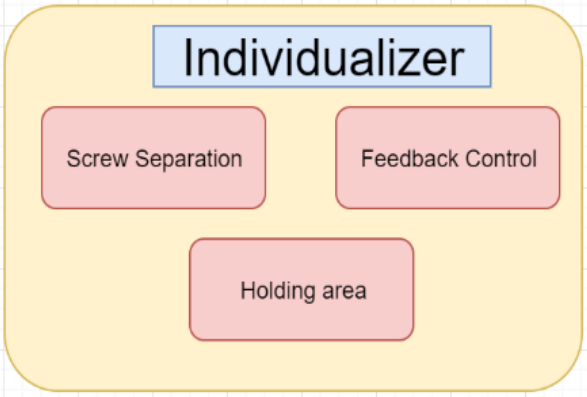
- Hold up to 100-200 unsorted screws
  - limit size of holding container mechanism
- Identify screws accurately
- Sort screws with high accuracy in a reasonable amount of time
- Complete sorting as fast as possible
  - Speed not high priority
  - Accuracy is key

Block Diagram

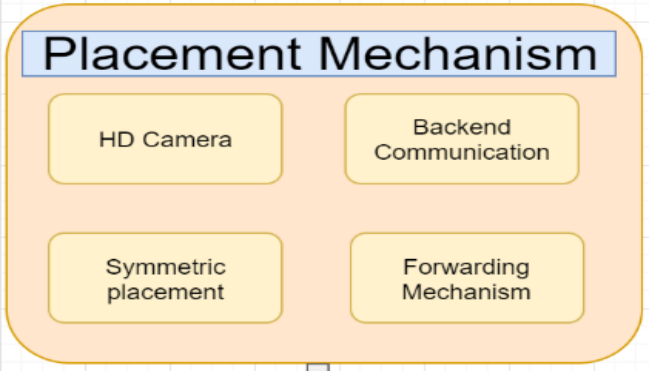


Individualizer

- Requirements
  - Able to separate and move screws into loading position one at a time.
  - Takes input from backend system letting it know when to stop and resume loading screws.
    - Response time under 100 ms
- Implementation
  - Vibratory bowl feeder
    - Can place a pile of screws into a bowl which uses vibrations to move screws one by one into a position to be scanned accurately



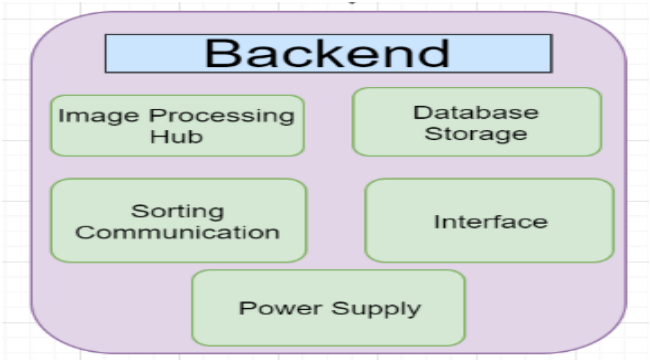
Placement Mechanism



- Camera will take an image of the screw
  - HD Camera / Webcam
- Symmetric placement
  - All the screws are placed in same place of the camera
- Forwarding Mechanism
  - Moves screws to the sorting mechanism
- Backend Communication
  - Communicates with database

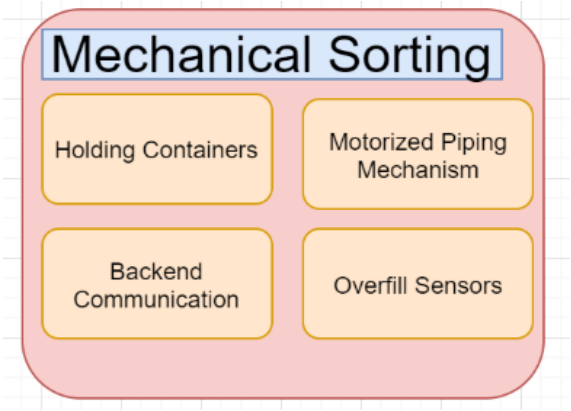
Backend - CPU and Image Processing

- Image processed via a Raspberry Pi
  - Running OpenCV Library w/ Python
  - Linux operating system
- Image database built as screws are processed
  - Screws compared to previous ones
  - Will account for 5-10 different screws



Mechanical Sorting

- Receive correct sorting location from backend
- Align containers to receive screws
  - Circular Shifting
  - Multiple Levels
- Measure level of container to prevent overfill
- Containers store sorted screws
  - 5-10 for separated screws
  - Single bin for miscellaneous screws to be resorted



System Summary

- Single screw separated from an assortment of of unsorted screws
- Screw is then placed in front of camera and scanned
- Image is taken to backend and compared to screws that have already been seen
- Backend deciphers where screw should be and sends assessment to sorting mechanism
- Screw is moved into the correct sorted bin

Team Roles (Technical)

- Hardware Leads: Jordan Gyaltsen & Rajesh Shahi
  - Identify solutions to mechanical problems
  - Implement cost-effectively
- Backend Leads: Andrew McGrath & Jordon Balskus
  - Decipher computerized system to handle sorting
  - Create real-time communication with mechanisms

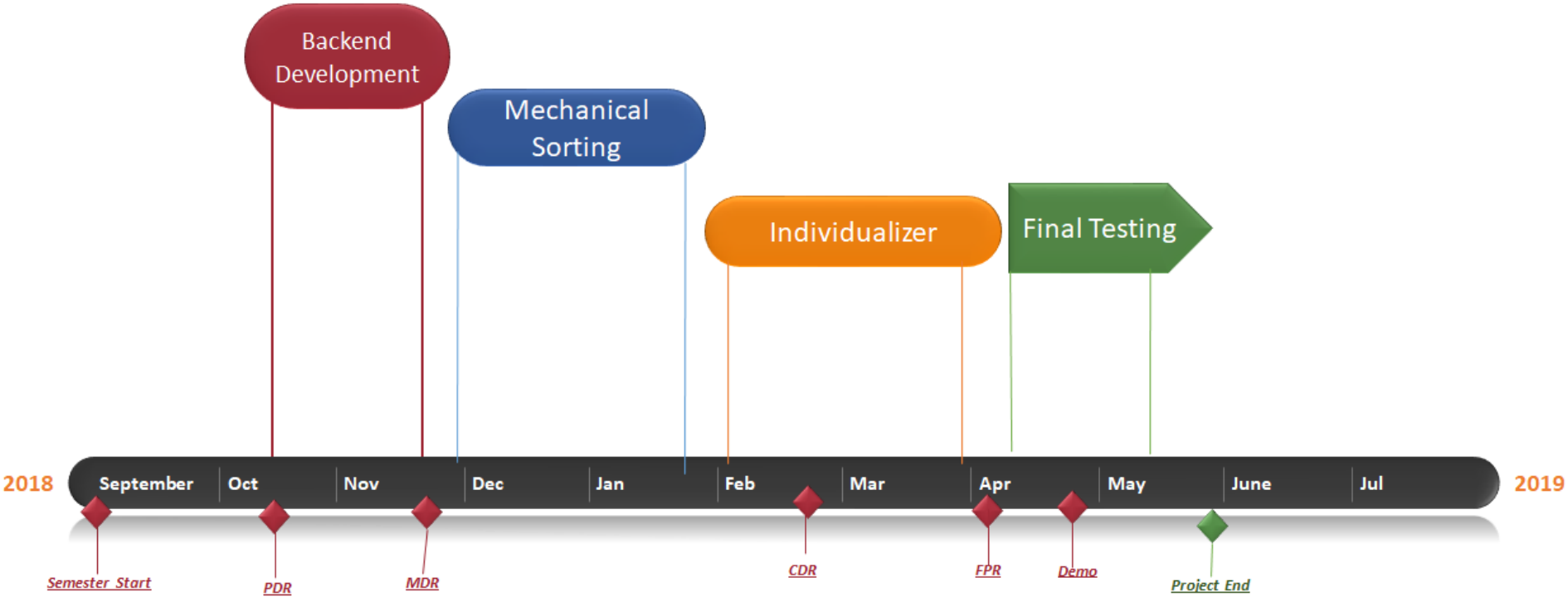
Team Roles (Administrative)

- Communication Manager: Rajesh Shahi
  - Handle communication with outside resources
- Technical Manager: Jordan Gyaltsen
  - Ensure technical components are being completed as planned
- Team Manager : Andrew McGrath
  - Schedule meeting times & coordinate reports
- Financial Manager: Jordon Balskus
  - Devise foreseen budget & handle ordering parts

Prototype Budget

- Total - $465
- Testing Screws – $20
- Enclosure Materials – $125
- Primary Sorting Motors/Mechanical Devices - $200
- Camera - $40
- Raspberry Pi - $40
- All image processing software – FREE
- Interface
  - LCD Screen - $40

Project Timeline



Proposed MDR Deliverables

- Demonstration of preliminary image processing
  - Have basic system that takes image of screw and imports it to our software suite
  - Demonstrate that we are able to detect screw length and diameter
- Demonstration of final screw placement
  - After being scanned and identified, a few screws will be placed into a designated bins.
  - Demonstrate the working of the mechanical system to complete sorting process.

Questions? Comments? Concerns?