# IntelliStorage (D3)

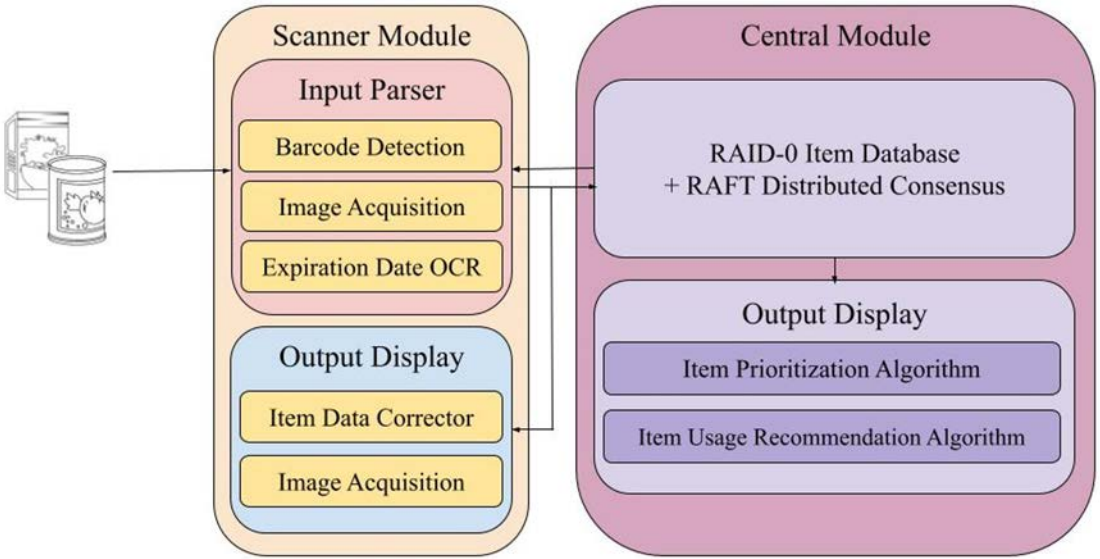
Jason Kim, Siyuan Li, Yuma Matsuoka

# Use case

## Provide a convenient method of keeping track of groceries at home

| Requirement #1<br>(Item Registration & Tracking) | Requirement #2<br>(Scaling) | Requirement #3<br>(Accessibility/Ease of Use) |
|---|---|---|
| **>90%** read-in accuracy | **40 items** per storage space | Store information within **1 sec** of scanning |
| **30 degree** scanning angle | **3 storage spaces** per network | Display info within **500 ms** |
| **15-25 cm** scanning distance | **10 sec** Synchronization | Daily report of expiring item |
| **4 sec** registration time | Data integrity | **5 min** setup node |

# Solution Block Diagram



# Complete Solution (Hardware)

- WoneNice USB Laser Barcode Scanner
  - Barcode acquisition
- NexiGo N60 1080P Webcam
  - Expiration Date acquisition
- FREENOVE 5 Inch Touchscreen Monitor (central module)
  - User Interface
- iPistBit 7 Inch Touchscreen Monitor (scanner module)
  - User Interface, Manual Overrides
- Raspberry Pi 4/5
  - Logic Unit for Scanner/Central Computer Modules

# Complete Solution (Software)


Boundingbox: 2021.08.03

Date Format: YYYYMMDD

Exp Date: 2021/08/03

```
Scan the Barcode:
016000335301
barcode: 016000335301
name: Nature Valley Crunchy Granola Bars  Oats  N Honey  18 ct  26.82 oz
brand: Nature Valley

Is this Item correct?

Y or N:
Y
Item selected
Scan the Barcode:
Sent: {'ScanDate': datetime.date(2024, 4, 21), 'Itemupc': '016000335301', 'ItemN
ame': 'Nature Valley Crunchy Granola Bars  Oats  N Honey  18 ct  26.82 oz', 'Ite
mBrand': 'Nature Valley', 'MessageType': 'REGISTER', 'messageID': 695873, 'ID':
222366147360316}
```

Item in Database

```
Scan the Barcode:
27983478
Not found in Database

Item not found. Please manually enter its information
ItemUPC (barcode number) (enter 0 if none):

0
Item Name:

Potato
Item Brand (enter NA if none):

NA
```

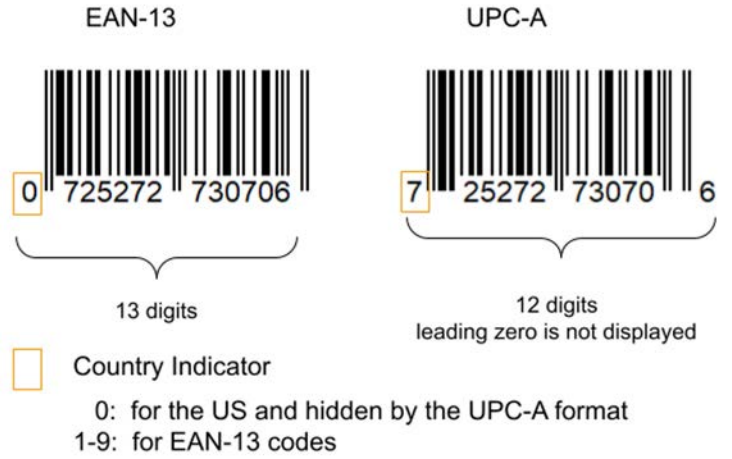
Item not found

# Design Trade Offs

- Human Interaction
  - Ease of Use (larger touch screen) vs cost of system (accessibility)
  - Chose 7” for scanner module, 5” for central computer (based on usage)
- Camera Focusing Time
  - Quality of Image (blur, light saturation) vs cost of system
  - Chose a cheaper camera with slower focus speed, more work on OCR algorithm
- Message Broadcast Intervals
  - Fresher Data vs Idle Computational Load
  - Chose 1 minute broadcast during idle (as heartbeat), 4 second broadcast during active (non-negligible workload)

# Design Limitation

- Item Regionality (Mainly US Products)
  - Limited to what is stored in UPC Item Database (US standard)
  - Limited EAN codes (international standard)
    - Very hard to find comprehensive EAN database





4908012001489
(EAN - Japan) ❌


8996212800332
(EAN - Indonesia) ❌


04142006781
(UPC - US) ✅

Image Source: https://support.scandit.com/hc/en-us/articles/4405347008914-EAN-13-vs-UPC-A-and-the-number-of-digits

# Testing, Verification, and Metrics 1

Temperature Sensor Accuracy

- Expected: **±2** ℃ of relative temperature, **±3%** of relative humidity.
- Actual: **±1** ℃ of relative temperature**, ±2%** of relative humidity. ✅

Recommendation Algorithm Testing

- Expected: In order results on boundary conditions (100% Exp Date, 0% Scan Date, etc)
- Actual: In order results, sensical output ✅


# Testing, Verification, and Metrics 2

1. **Single** Module Testing – Item Registration
- Mimic use case. Continuously scan items in manner that tests parameter boundary. Use items found at home. Scan items varying in shape and sizes.
- ✅ if observed values hit design requirement target.

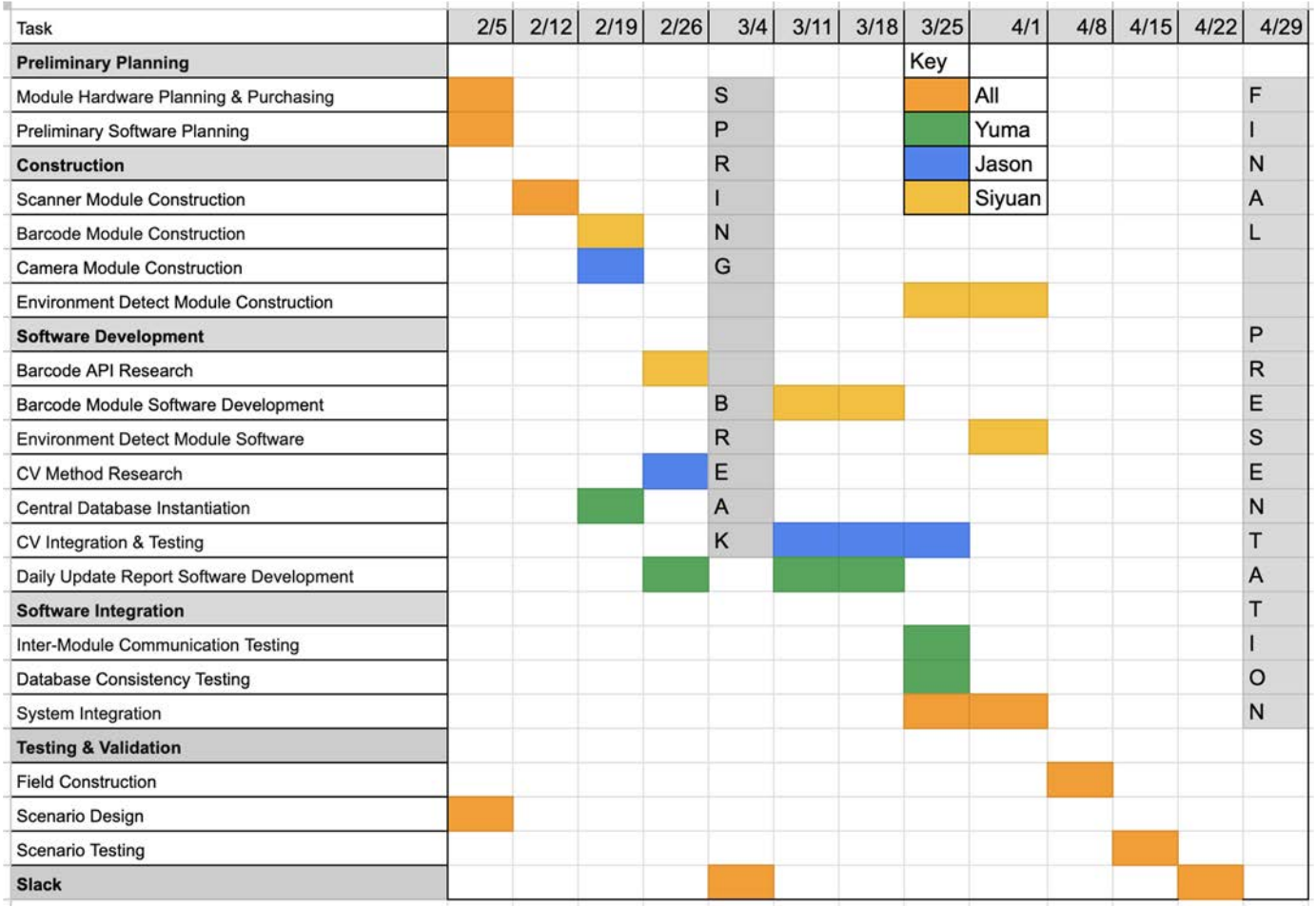
1. **Multi**-Module Testing – Scaling & System Reliability
- Test overall system by scaling up single module test to three modules. Simulate crash by shutting down nodes and ensure data restoration & shared consensus afterwards.
- ✅ if scaling meet threshold values; synchronization, data consistency, are maintained.

1. **Usability** Testing – UI Ease of Use
- Test UI lag and ease of use on actual hardware
- ✅ if intuitive to use (subjective) and hits use case lag requirement (quantitative).

# Testing Summary

| Requirement #1<br>(Item Registration & Tracking) | Requirement #2<br>(Scaling) | Requirement #3<br>(Ease/Accessibility of Use) |
|---|---|---|
| Target: **>90%** read-in accuracy<br>Result: **73%** accuracy<br>Cause: Blurry saturated photos from shaky hand ❌ | Target: **40 items** per storage space<br>Result: **>100 items** ✅ | Target: Store information within **1 sec** of scanning<br>Result: **100ms** ✅ |
| Target: **30 degree** scanning angle<br>Result: max **35 degrees** ✅ | Target: **3 storage spaces** per network<br>Result: ✅ | Target: Display info within **500 ms**<br>Result: **In progress** 🔺 |
| Target: **15-25 cm** scanning distance<br>Actual: **5-30 cm** ✅ | Target: **10 sec** Synchronization<br>Result: **500ms** ✅ | Target: Daily report of expiring item<br>Result: **In progress** 🔺 |
| Target: **4 sec** registration time<br>Actual: **3 sec** ✅ | Target: Data consistency<br>Result: ✅ | Target: **<5 min** setup node<br>Result: **4 min** ✅ |

# Schedule


Task
2/5
2/12
2/19
2/26
3/4
3/11
3/18
3/25
4/1
4/8
4/15
4/22
4/29
Preliminary Planning
Module Hardware Planning & Purchasing
Preliminary Software Planning
Construction
Scanner Module Construction
Barcode Module Construction
Camera Module Construction
Environment Detect Module Construction
Software Development
Barcode API Research
Barcode Module Software Development
Environment Detect Module Software
CV Method Research
Central Database Instantiation
CV Integration & Testing
Daily Update Report Software Development
Software Integration
Inter-Module Communication Testing
Database Consistency Testing
System Integration
Testing & Validation
Field Construction
Scenario Design
Scenario Testing
Slack
SPRINGBREAK
PRESENTATION
Key
All
Yuma
Jason
Siyuan

# Conclusion

## Challenges

- UI integration
- OCR development

## Takeaways

- Integration takes *time*
- Don’t try to *overachieve*
- Considering and dealing with edge cases is *hard*


Image Source: https://kitchencabinetkings.com/glossary/pantry/