

**In-House Packing Engine for  
MARSHALLTOWN**

# MEET THE TEAM

Evelyn Smith - Project Lead  
Computer Science Major  
with a Mathematic Minor

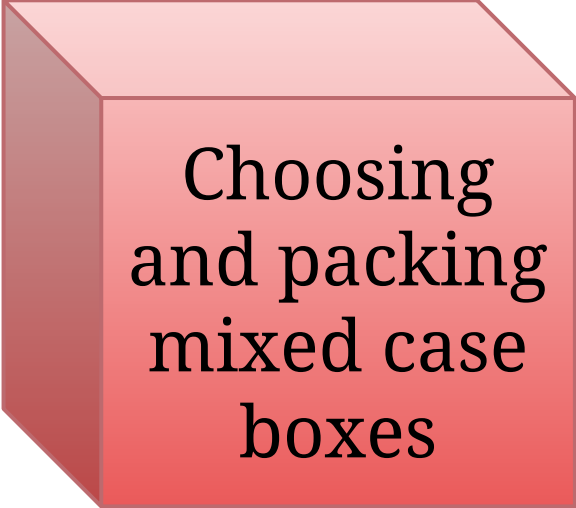
Carey Lawrence - Team Member  
Computer Science Major  
with a Mathematic Minor

Akhila Parvathaneni - Team Member  
Computer Science Major  
with a Bachelor's in Biology

# Key Personnel

- Craig Wall
  - Director of Arkansas branch IT
- Jeff Schneider:
  - Director of MARSHALLTOWN IT across all branches
- Steve Smith
  - Main source at MARSHALLTOWN for any SQL scripts and database management

# Problem

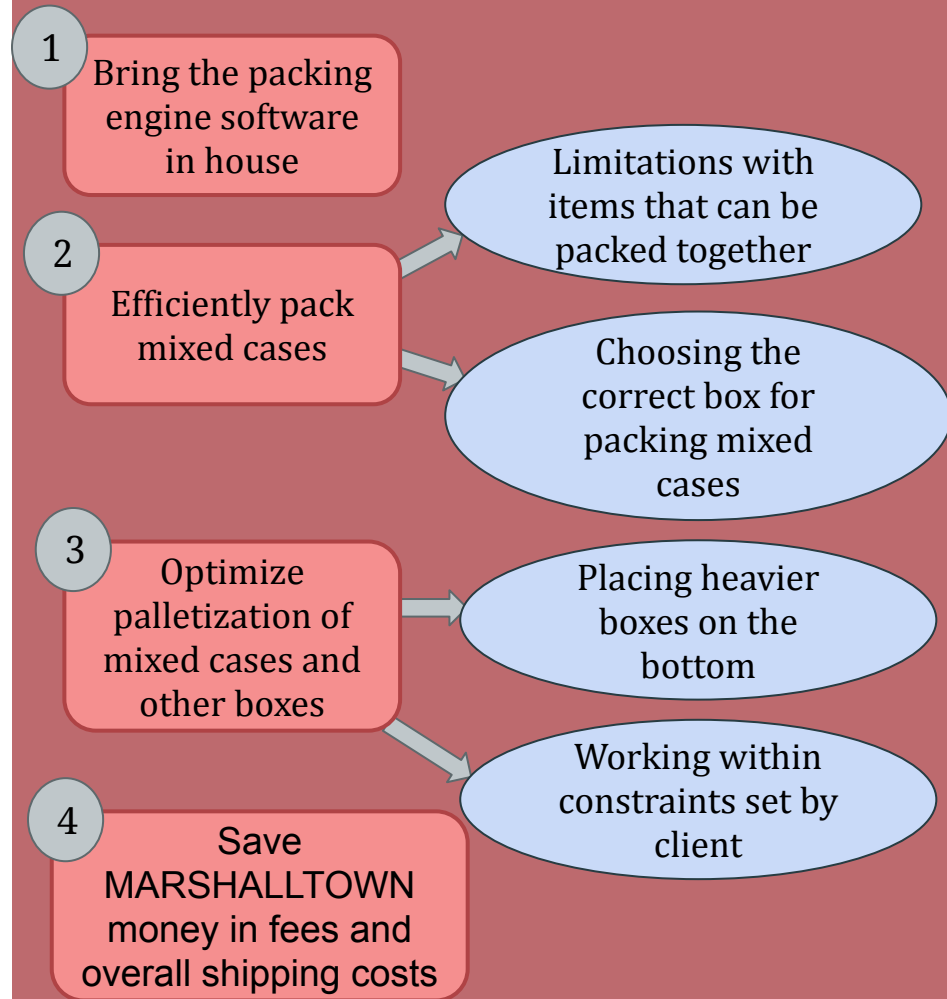


Choosing  
and packing  
mixed case  
boxes



Palletizing  
boxes  
efficiently

# Objectives



# Background

Preexisting Code:

Current packing engine

Starter project-- EB-AFIT  
algorithm

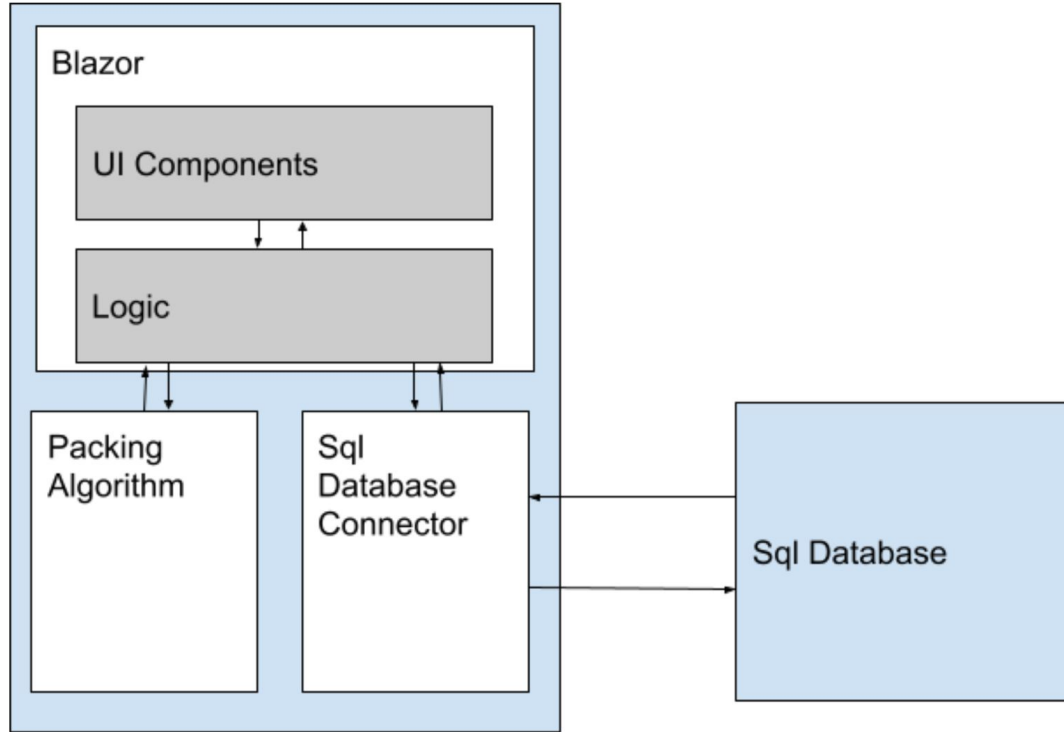
- ❖ C#
  - Blazor client side
    - .NET Core
  - C# Server side
    - .NET Standard
      - Algorithm
      - Linq2Db
- ❖ SQL Database
  - Stored Procedures
  - Box information

# Design

## Requirements

- ❖ For Mixed cases...
    - Most cost efficient box size, weight, and quantity of boxes for a given list of items to be packed
  - ❖ For pallets...
    - Prioritize heaviest items at the bottom
    - Certain items cannot be reoriented
    - Abnormal items pose large issues during packing
    - Packing items of the same type together
  - ❖ Overall...
    - Maximize cost savings for MARSHALLTOWN shipments
-

# Design - Architecture





# Design

## Database

- ❖ Base schemas from MARSHALLTOWN
  - ❖ Hosted on Azure
  - ❖ Two tables used
    - ItemsToPalletize
    - BoxData
  - ❖ Alterations made
    - CanBeFlagpole
    - CharWeight
    - IsAbnormalPack
  - ❖ Test Datasets
    - R44999 (general test set), R55999 (items grouped together), R44888 (minimal set)
-

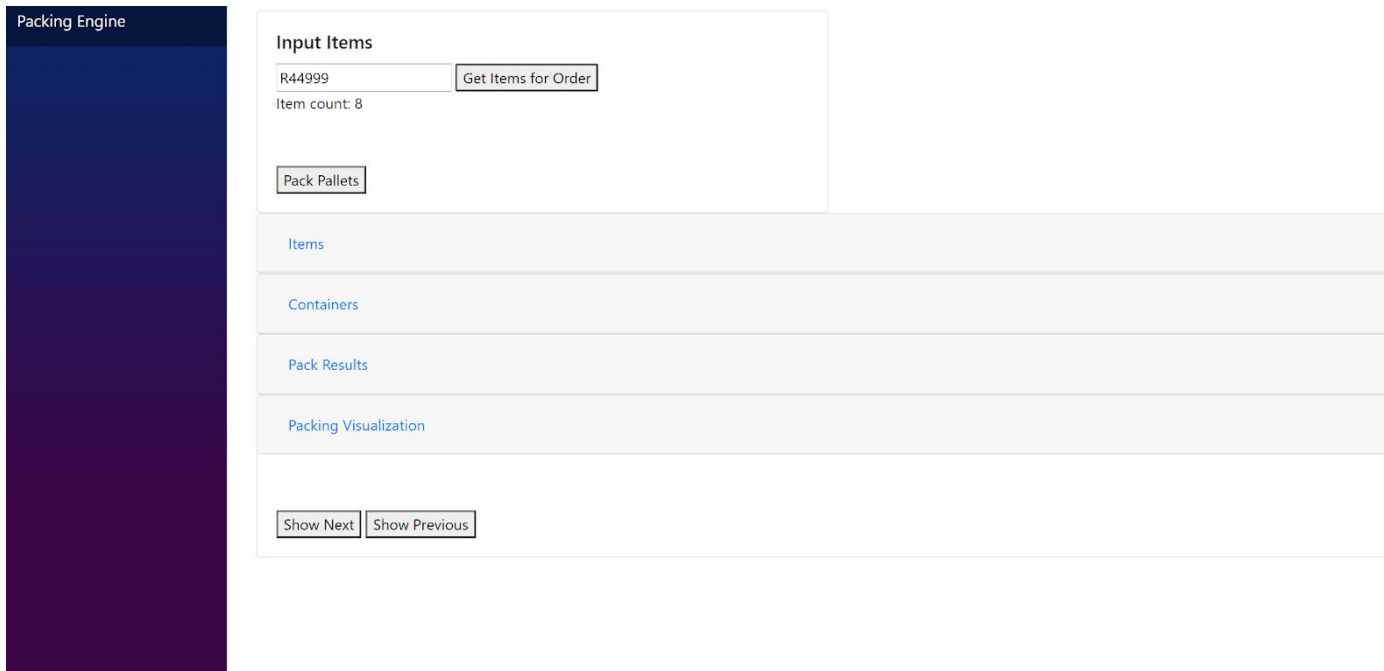
# Design

## Database Connector

- ❖ Utilizes Linq2DB
    - Put connection information into .tt file and save
    - Generates class with database information
    - Uses C# Interface that performs the actual retrieval to access data in other projects
-

# Design - Frontend

## Blazor with THREE.js



## Items

Name	L	W	H	Qty	Weight	Can be packed flagpole?
28167c6	14.30	5.70	8.10	10	B	Yes
28168c6	14.60	6.50	8.20	8	C	No
28169c6	9.00	1.40	6.40	10	C	Yes
28169p576	48.00	14.10	40.00	2	A	Yes
28170c6	9.80	1.70	6.20	10	C	No
28170p504	48.00	13.00	40.00	1	B	Yes
28171c6	9.20	2.50	6.20	10	C	Yes
28172c6	9.00	1.20	5.10	10	C	No

## Containers

Name	L	W	H	Volume
WS1075	37	11	9.3	3785.1
SN3300	21	14	16	4704
SN3433	31	16	11.5	5704.0
WS390	17	13.625	15.5	3590.1875
WS391	21	11	12.25	2829.75
SN3434	39	20	11.5	8970.0
WS399F	15.25	8.75	12	1601.2500
WS432	14.125	8.25	5	582.65625

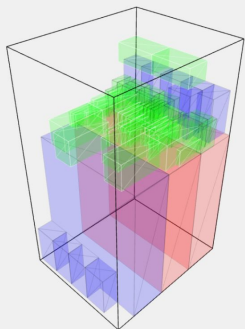
## Pack Results

Algorithm Name	Pack Time (ms)	% Cont. Used	# Items Packed	# Items Unpacked
EB-AFIT	19	69.26	61	0

0

Show Next

Show Previous



## THREE.js Visualization Key

- ❖ Red Box: Weight Class A
- ❖ Blue Box: Weight Class B
- ❖ Green Box: Weight Class C
- ❖ Black Outline: No orientation restrictions
- ❖ White Outline: Vertical orientation restrictions

Algorithm Name	Pack Time (sec)	% Cont. Used	# Items Packed	# Items Unpacked
Skull	1	99.99	41	0

Packing Visualization



Show Next Show Previous



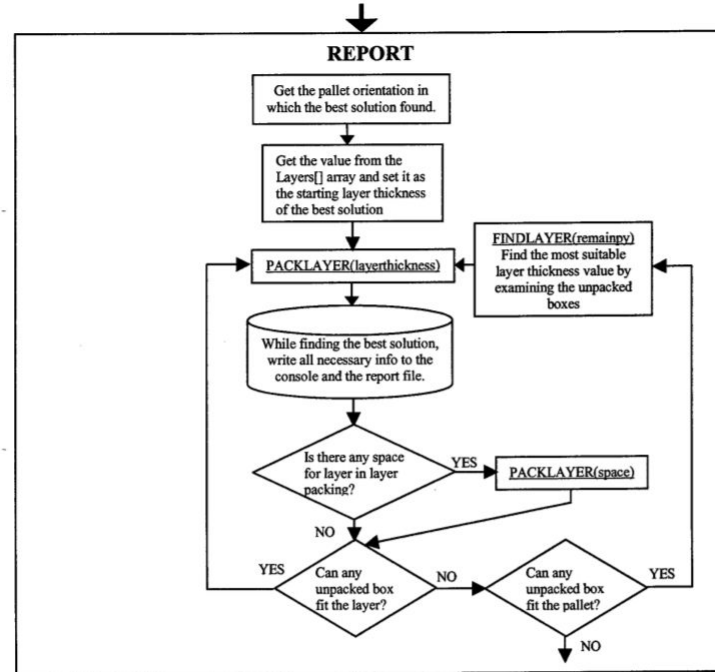
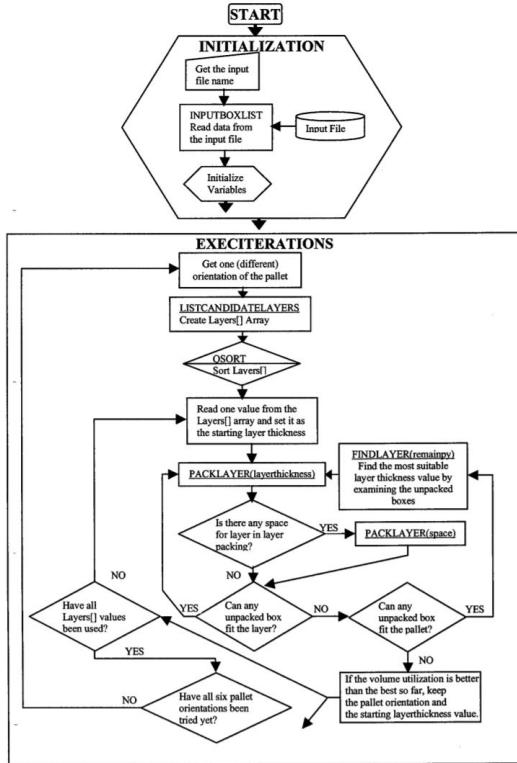
00:00:14

# Design

## Algorithm

- ❖ Utilizes EB-AFIT Algorithm
    - Algorithm from a master's thesis in early 2000s
    - Designed to mimic human thought while efficiently packing a pallet
    - Approximately 1500 lines
-

# Design - Algorithm flow





# Design - Grouping Items

Changes made outside of algorithm and effects algorithm entry

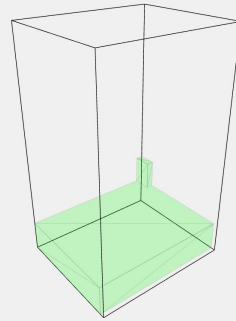
Tries to pack by item, if a layer has been made then group by layer and record qty to make a layer with item

Packing Visualization

0

Show Next

Show Previous



# Design - Restricting Orientation

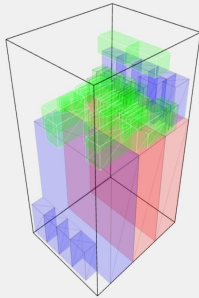
Added respective bit to flag if an item can be reoriented

Determined that the FindBox method was where check for orientation should occur

Packing Visualization

0

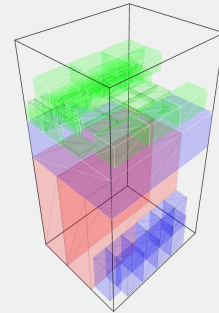
Show Next Show Previous



Packing Visualization

0

Show Next Show Previous



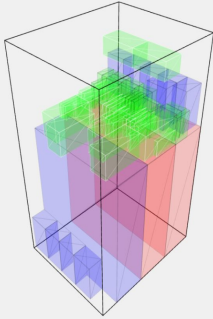
# Design - Weighted Item Stacking

Broke down list of items to be packed within the EB-AFIT algorithm to three Lists, pertaining to each weight class

Packing Visualization

0

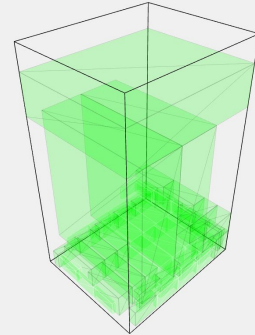
Show Next Show Previous



Packing Visualization

0

Show Next Show Previous



# Design

Ongoing Development

- ❖ **Abnormal Items**
    - Will utilize disallowing of rotation
    - In FindBox, set height of item to max remaining height in pallet and push initial item height to a list
      - Reset item's height for visualization during OutputBoxList
  - ❖ **Container Packing**
    - Will occur outside of EB-AFIT algorithm
    - Will loop through containers with items and when a volume metric is met, save the container as successfully packed and remove those items from the list
-

# Tasks

1. Determine and create script for SQL Database creation

2. Learning new software and frameworks

3. Testing current project and determining any additional shortcomings

4. Algorithm return model determination/updating

5. Adjusting the algorithm for heaviest at the bottom

6. Adjusting the algorithm to work with items that cannot be reoriented

7. Adjusting the algorithm to work with abnormally shaped items

8. Adjusting algorithm to determine most efficient box size for given list

9. Adjusting algorithm to determine most efficient box weight for given list

10. Adjusting algorithm to determine most efficient box quantity for given list

11. Making packing algorithm work for both pallet packing, and box packing...

12. UI and visualization rework

13. Unit testing application

14. Edge case testing application

# Future Work

- ❖ Abnormal Item Orientation
  - ❖ Container Packing
  - ❖ Better UI prompts
    - Show current Item/Container name when packing
    - If a layer of one item, show count of them and item type
  - ❖ Grouped Item fixes
    - Make sure the layer of grouped items won't fall off overhang
  - ❖ Get customer's pallet requirements by custOrderNo (would require more MARSHALLTOWN data)
-

# Facilities and Equipment

- C# pallet packing
    - Blazor
    - SQL Database connector
    - Packing algorithm
  - SQL scripts
-

Questions?