



**University of Arkansas – CSCE Department
Capstone II – Preliminary Proposal – Fall 2021**

In-House Packing Engine for MARSHALLTOWN

Carey Lawrence, Akhila Parvathaneni, and Evelyn Smith

Abstract

In this paper we will outline the in-house packing engine for MARSHALLTOWN to improve spending costs on shipping. This packing engine will optimize packaging smaller items in the most appropriate larger box as well as the stacking of all items on a pallet. This packing engine will take into account individual box dimensions, weight, space limitations from the client and shipping company and the vast array of potential solutions. In order to achieve this, we are starting with a space optimization program and test databases provided by MARSHALLTOWN.

Our approach to this is to start with the front-end development to get the user interface working correct. The next step will be to take each of the additional limitations of packing. Moving this software in-house will provide many benefits. The first benefit of our project is creating a pallet optimization that will take into account weight instead of just box dimensions. Additionally, MARSHALLTOWN will have the ability to adapt the program as needed and add new features as new limitations present themselves. Lastly, it will help pack the same amount of area more efficiently and therefore reduce overall shipping costs.

1.0 Problem

MARSHALLTOWN needs a packing engine that provides the warehouse workers the best options for packing items requested by the client. Before going any further, let us define what we mean by “packing engine”. For MARSHALLTOWN the packing engine does two things. The first problem is finding the best way to pack the smaller items into a larger, more portable box in order to save money on shipping costs, but not too large of a box that consumes more airspace than needed. The second half of the problem is taking the larger packed boxes and larger items in general to pack them onto a pallet of which has size limitations specified on the client’s end.

As of now, they have a packing engine already in place. The issue with this is that the service is outsourced from another company. Meaning that they are not really sure how the program is working. MARSHALLTOWN would like to move the packing engine in house to have a better idea of how the code is working, possibly make it more efficient and to add on other requirements as needed.

2.0 Objective

This project has multiple objectives. The first being to create a packing engine that takes smaller items and the different available box sizes to find the most efficient way to pack the smaller

items. In this case efficient takes on the meaning of containing the least amount of open space in a box. We want to fill the box as full as possible with the items given and choose the appropriate box to pack in. The next objective of this project is to efficiently palletize all of the requested products within the height and weight limitations of the client and also the delivery service. This is done to minimize additional fees from both the client and delivery service as well as take down the entire cost of delivery by taking up less space in general.

3.0 Background

3.1 Key Concepts

Some key technologies that are relevant to this project are C#, .NET Standard, Blazor, and SQL. C# is an object-oriented language developed by Microsoft that has similarities to Java. C# enables users to develop Web Applications. The .NET Standard framework is multi-platform which allows for other languages such as C# to use it. .NET can provide services specific to building the application such as accessing the time. There is another tool known as ASP.NET that allows the user to develop web programs. We will be using .NET as the backend for our project.

For the front-end, we will be using Blazor which is an open-source framework that allows the user to build interactive web applications using C#. The name itself combines “Browser” and “Razor” and Blazor supports the performance of the client-side views. More about how Blazor is used specifically for our project can be found in a later section of this report.

Another language that is relevant to our project is SQL. SQL or Structured-Query Language is language used in programming to establish communication with the database. The SQL statements are used to obtain specific pieces of data from the database using certain conditions. SQL is also used to update or modify the database depending on the situation. In general for this project we will be connecting to the SQL database using linq2db library from .NET to evaluate data.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

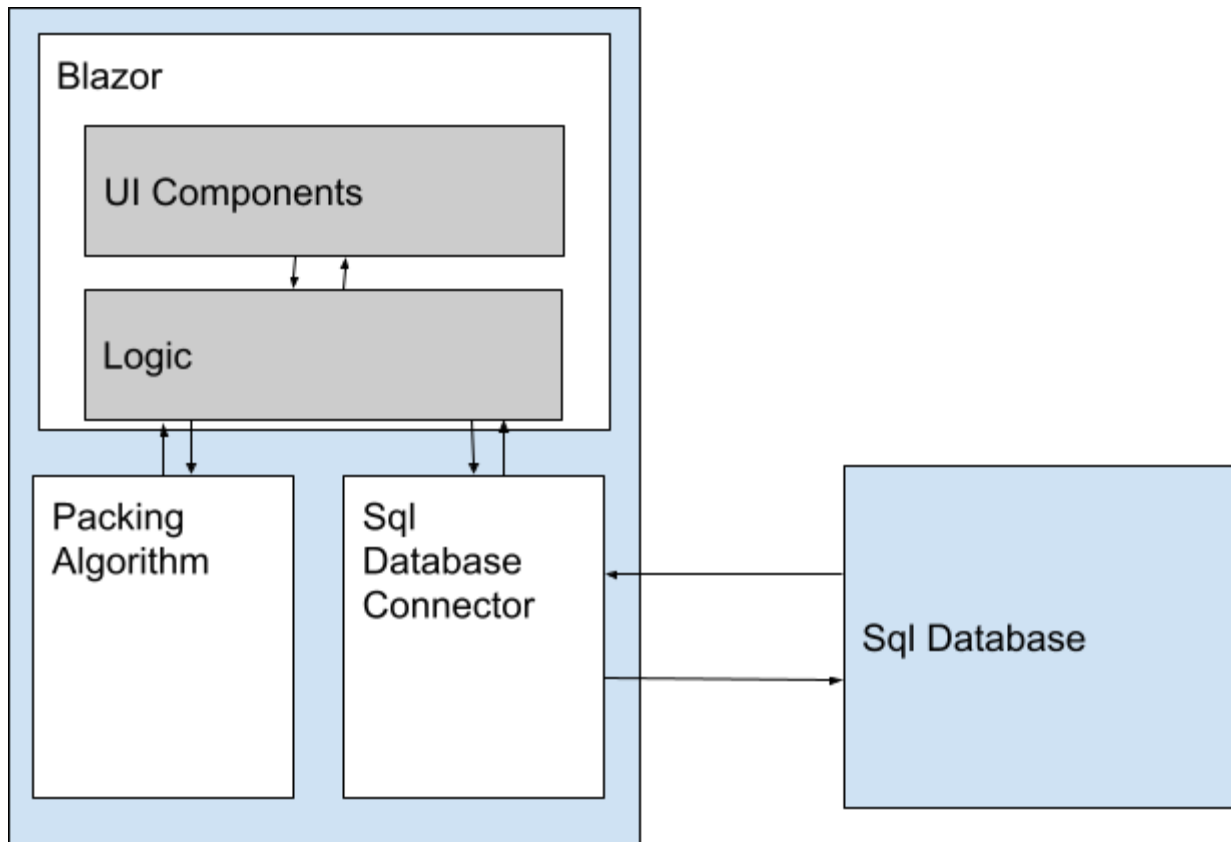
As mentioned, there are two different jobs that this application will serve. The first is selecting appropriately sized boxes to pack mixed cases and how said boxes will be packed. The second will be to determine the most efficient way to pack a pallet given its dimensions and what’s to be packed on said pallet. The front-end requirement for this project will be to create an interface that works intuitively as possible to provide the user with an easy way to enter any items, boxes, or dimensions that will restrict the order.

Because of the products that MARSHALLTOWN produces, many additional requirements will factor into the algorithm we will be designing. At MARSHALLTOWN, many products ship in case quantities or are mixed cases with many different items inside. A requirement of this algorithm is to prioritize heaviest cases at the bottom. Additionally, many products must be

shipped following a certain orientation, therefore the algorithm may not be allowed to alter the orientation of certain items, even at the cost of efficiency. Finally, MARSHALLTOWN products such as industrial rakes have abnormal, triangle shaped boxes and are packed with an exposed handle with a length of around 60". These requirements cause great difficulty when packing and are what makes this project with MARSHALLTOWN so unique.

4.2 High Level Architecture

For the front-end design of our webapp, we plan to use the Blazor framework. Blazor is an extension of the .NET developer platform that allows for the creation of Razor components. These components then allow for the integration of HTML and C# code to create web pages. Unlike MVC apps, Blazor does not follow a request/response model, but instead centers around client-side UI logic where the logic can then make further calls to the .NET Standard backend [1]. This UI is meant to be intuitive to enter data into, and offer clear explanations of how the results of a call to the packing algorithm should be interpreted.



This project will utilize a sql database of sample data used at MARSHALLTOWN. To access, manipulate, and create the stored procedures for the database, we will be using SQL Server Express as our software of choice.

Our connector to the SQL database will utilize the .NET Library linq2db to send and retrieve the data to and from our SQL database. The data acquired through this method will be used to validate items and box sizes, as well as to retrieve information regarding specific customers' needs for pallet packing.

The final large component of our project is the packing algorithm. While we have been given a starting C# solution for this project, we will find the majority of the changes in this project being made in this part of our work. The packing algorithm is what this project largely revolves around, with the end goal of this specific component being to accurately compute the most cost efficient way to pack boxes and pallets. This component will communicate with our Blazor project to take in the data regarding what is to be packed, and then to return a model of how those items should be packed.

4.3 Tasks

As this is still in the earliest phases of development, the tasks listed are likely more general than what will end up becoming of them as we progress.

1. Determine and create script for SQL Database creation
2. Learning new software and frameworks
3. Algorithm return model determination/updating
4. Testing current project and determining any additional shortcomings
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, with individual methods for each
12. UI and visualization rework
13. Unit testing application
14. Edge case testing application
15. Bug documentation/Ongoing process documentation (ongoing)

4.4 Schedule

Tasks	Dates
1. Determine and create script for SQL Database creation	9/13-9/17
2. Learning new software and frameworks	9/13-9/20
3. Testing current project and determining any additional shortcomings	9/20-9/22
4. Algorithm return model determination/updating	9/22-9/24
5. Adjusting the algorithm for heaviest at the bottom	9/24-10/1
6. Adjusting the algorithm to work with items that cannot be reoriented	10/1-10/8
7. Adjusting the algorithm to work with abnormally shaped items	10/8-10/15

8. Adjusting algorithm to determine most efficient box size for given list	10/15-10/22
9. Adjusting algorithm to determine most efficient box weight for given list	10/22-10/29
10. Adjusting algorithm to determine most efficient box quantity for given list	10/29-11/5
11. Making packing algorithm work for both pallet packing, and box packing...	11/5-11/12
12. UI and visualization rework	11/12-11/19
13. Unit testing application	11/19-11/26
14. Edge case testing application	11/26-12/3

4.5 Deliverables

For this project, we will have two main coded deliverables:

1. C# Pallet Packing solution containing 4 projects...
 - a. Blazor front-end with .html layouts, .razor components, startup settings, and other classes used directly from the Blazor frontend
 - b. SQL Database connector using .NET's linq2db library with methods to get and map the data returned from our SQL calls
 - c. Packing Algorithm project with all necessary C# files
 - d. Unit testing project with predefined tests to rapidly show the accuracy of our algorithm
2. SQL Database scripts
 - a. Script for setting up the database we will be working with
 - b. Script for populating the database
 - c. Script(s) for creating any stored procedures we may use

Another deliverable for this project will be the design document that reflects all resources, design decisions, and any further design-related documentation.

Finally, we will have the final report to write and turn in that details the work we've completed on this project throughout the semester.

5.0 Key Personnel

Evelyn Smith – Smith is a senior Computer Science major in the Computer Science and Computer Engineering Department with a minor in Mathematics at the University of Arkansas. She has taken Database Management, Programming Paradigms, Software Engineering, and is currently enrolled in Algorithms. Smith has been working with MARSHALLTOWN since June of 2021 and has had the opportunity to work with all of the frameworks and libraries used in this project in a professional setting. Responsible for SQL Database and Blazor UI.

Carey Lawrence – Lawrence is a senior Computer Science major in the Computer Science and Computer Engineering Department with a minor in Mathematics at the University of Arkansas. She has completed relevant courses including: Algorithms, Database Management, Programming Paradigms, and Software Engineering. Additionally, Lawrence has done research into machine

learning as well as full-stack development for multiple web based applications through Credera Consulting. Responsible for developing our algorithm to work with pallet restrictions.

Akhila Parvathaneni – Parvathaneni is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. Parvathaneni has already obtained a Bachelor Degree in Biology and a minor in Mathematics. She has completed relevant courses including: Artificial Intelligence, Algorithms, Software Engineering, and Programming Paradigms. She is familiar and has experience with iOS/Android Development and Full-Stack development. Responsible for developing our algorithm to work with mixed case restrictions.

Craig Wall, Industry champion – Wall did not provide us with a short biography by the time that this was due.

Jeff Schnieder, Industry champion/IT Director – Schnieder has been developing software professionally for 20 years and previously owned his own company for 8 years before joining MARSHALLTOWN in 2010. Schnieder manages the organization's development and systems teams. We have developed many novel software products for our own use and in 2014 started selling some to other distributors/manufacturers. Since then we have productized a SDK that makes interaction between C# software and ERP systems, mainly Microsoft Dynamics AX based systems, much easier to write. Now we have a software company component in our department and we have many customers in the US and Europe.

1.0 Facilities and Equipment

The facilities and equipment used for our project as mentioned earlier involve the C# code with the packing algorithm, SQL Database connector, and Blazor for the front end.

7.0 References

[1] Architecture comparison of ASP.NET Web Forms and Blazor, <https://docs.microsoft.com/en-us/dotnet/architecture/blazor-for-web-forms-developers/architecture-comparison>

[2] linq2db - Introduction, <https://github.com/linq2db/linq2db/wiki/Introduction>