MATTRESS WAREHOUSE INVENTORY MANAGEMENT SYSTEM

Final Report

Yamall Velez

April 26, 2022

Hillsborough Community College

COP-2939-61167

Table of Contents

Project Proposal

Introduction

I have been working part time for a mattress warehouse as a picker. My job involves using a list, called a route, to grab various items around the warehouse and transport them to specific locations for the truck drivers to take away the next day. I have noticed that the system has its limitations, specifically regarding the accuracy of the location of stored items. As of now, the location of items are simply labeled with an aisle number. Items could be located anywhere down the length and height of each aisle.

This software is aimed to be used by both pickers and routers (employees who create routes). This project intends to create a virtual environment in which rules for where inventory should be stored. The intended benefits are to minimize the misplacement of items, create a more organized warehouse, provide pickers more information on an items whereabouts, and allow routers to create routes without the need to personally remember inventory item locations.

Development Properties

The software is intended for use on Windows systems. The software will be written in C# with the Visual Studio 2019 development environment.

Project in Detail

User Interface **–** The software's user interface will be created with the intent of streamlining the use of the software's features and capabilities. The use of textboxes, checkboxes, dropdown lists, buttons, and menus should provide a convenient way for the user to properly utilize this inventory system software.

Modular Storage Design **–** The warehouse storage currently consists of numbered aisles. These aisles can be further divided into columns and rows. Columns are sections that represent how "long", and aisle is, while rows are sections that represent an aisle's "height". Where and a column intersect, a cubicle is formed. These cubicles are where items are stored/ The software will allow the ability to add or remove aisles, columns, and rows. The software will also allow to specify the dimensions a cubicle, such as its length, width, and height. Altering a cubicle' dimension will also affect a row and column's dimensions. Aisles, rows, and columns will have the option to blacklist or whitelist items based on their attributes and classifications.

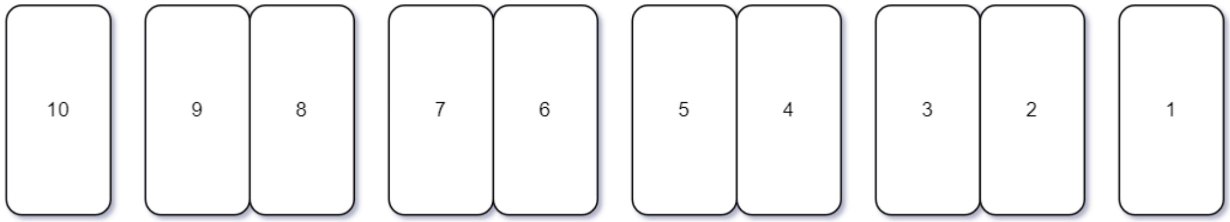


Figure 1 - Aisle top view. The space in between every 2 aisles is necessary for item retrieval

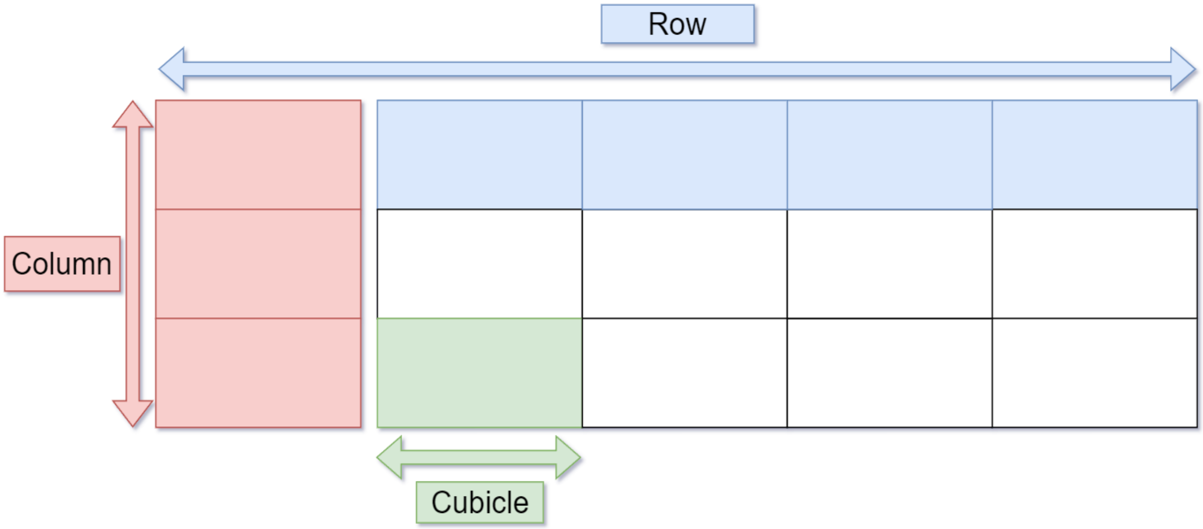


Figure 2 - Aisle side view. Aisles contain both columns and rows, which determine the number of cubicles, their height, and length. Cubicle depth is determined by aisle width.

Inventory Classification and Attributes **–** These classifications and attributes are additional information to provide more descriptive clarity for each item.

All inventory items will follow this general rule of attributes.

- Product Type – The type of product (i.e., mattress, box mattress, base, accessory, foundation)
- Manufacturer – The product’s manufacturer name
- Name – The product’s name
- ID – A number that is specific to the product model

Mattresses, bases, and foundations have two additional attributes.

- Size - The size of the mattresses based on the classification of Twin, TwinXL, Full, Queen, King, California King
- Dimensions – the width, length, and height of a mattress, Width will pertain to an item’s thickness.

Accessories Such as pillows, brackets, protectors, and frames contain the “pallet size” attribute.

- Pallet Size - The size of the pallet in which the accessories are stored on. This attribute gives an idea of the physical space a stored accessories will occupy based on a pallet’s width and length.

Adding and removing inventory **–**The user will be able to add and remove items from the inventory

Searching inventory **–** Inventory items will be able to be searched by their attributes and classifications. Their quantity and location will also be available upon search.

Potential Features **–** These are features that must be further evaluated

- Sort – The ability to sort items in the software. This would serve merely as a guide of how the warehouse could be sorted if such a task would be done physically.
- Additional attributes/descriptions – New attribute types could be created by the user of the software without changing the source code.
- Routing – Allowing the creation and editing of routes directly from the software. Routes are the list employees used to gather items and place them in a specified location.

Project Limitations

One of the first limitations of the project is the flexibility of the Modular Storage Design feature. All existing warehouses are not built the same, and the software can not factor in all physical configurations that exist in the world. Another limitation is that there may be items with unique attributes and storage necessities which do not fit any current item classification within the software. Creating a more flexible modular system would be a potential solution for this limitation.

Another limitation is my skills may not be at the level necessary for the project’s proposed features. Learning throughout the project’s development is a potential solution to this limitation.

Project Schedule

The project schedule will follow closely to the course outline created by Dr. Banisakher in our course syllabus. Throughout the project's deadlines I will create bit and pieces of the project's source code as to get familiar with what I've proposed and test my own capabilities. The idea is to analyze what I can do with my current skills, and what areas of the project I will need to allocate more time to research, study, and develop. Doing so will allow me to get a better understanding on an estimate of time I will spend per week working on this project.

Documentation

The software' documentation serves two purposes:

- To provide the user with the necessary information to operate the software efficiently. Documentation will include instructions on how to use the aforementioned features in the "Project in Detail" section.
- To provide other software developers a guide on the design choices and code structure of the software. Additionally, comments in the source code will provide guidance behind design choices.

Requirements

I. Introduction

I.1 Purpose of Document

This document is intended to list the functional and non-functional requirements for the Mattress Warehouse Inventory Management System. Additionally, this document will, in detail, guide through different user cases on the use of this software. As a whole, this document servers as a reference on the key points of the program and instructions on using it.

I.2 Intended Audience

This document is intended for pickers, routers, and any software maintenance staff that need a guide or reference for the inventory management software.

II. Functional Requirements

II.1 Modular Storage Design

Storage design overview needs flexibility to fit a wide range of warehouse floor spaces. Height, length, width, and number of aisles must be customizable.

II.2 Inventory Classification and Attributes

All inventory items have unique classifications that encompass a general type of item. All inventory have attributes that are shared across all classifications. Different classifications may contain additional unique attributes.

II.3 Adding and Removing Inventory

Inventory must be able to be created, added, or removed from active inventory.

II.4 Searching Inventory

Inventory can be searched to display its current location, or it's intended location if item has not been placed correctly. Inventory search is facilitated by modular storage design and Inventory classifications and attributes.

III. User Cases

III.1 Modular Storage Design

III.1a User adds the first aisle.

III.1b User adds an aisle to the left or right of an existing aisle.

III.1c User removes an aisle.

III.1d User sets parameter whitelist or blacklist for an aisle.

III.1e User adds or removes a row to an aisle.

III.1f User adds or removes a column to an aisle.

III.1g User sets parameter whitelist or blacklist for row, column, or cubicle.

III.2 Inventory Classification and Attributes

III.2a User classifies an inventory item.

III.2b User sets additional inventory item attributes.

III.3 Adding and Removing Inventory

III.3a User creates a new item.

III.3b User adds a new item to the inventory

III.3c User removes an item from inventory.

III.3d User moves an inventory item to a different location in the inventory.

III.4 Searching Inventory

III.4a User searches for an inventory item with a search bar and/or filter criteria.

III.4b User searches for an item on the Modular Storage Design view map.

III.4c User views all items in inventory.

IV. Non-Functional Requirements

IV.1 System Requirements

The project software is meant to be used on personal computers running Windows 10 or later. Program and associated files on disk will be small. Program and associated files, such as those saving the inventory will be stored locally on the device.

V. Potential Requirements

V.1 Inventory Sorting

Using the modular storage design, a virtual plan to sort out the warehouse inventory could be made possible. Sorting would be based on multiple factors.

- Classification
- Attributes
- Name

V.2 Custom User Added Classifications and Attributes

User will be able to create additional item classifications or item attributes as necessary. User will also be able to remove existing classifications and attributes.

VI. Limitations

VI.1 Modular storage design may not be flexible enough for specific warehouse configurations.

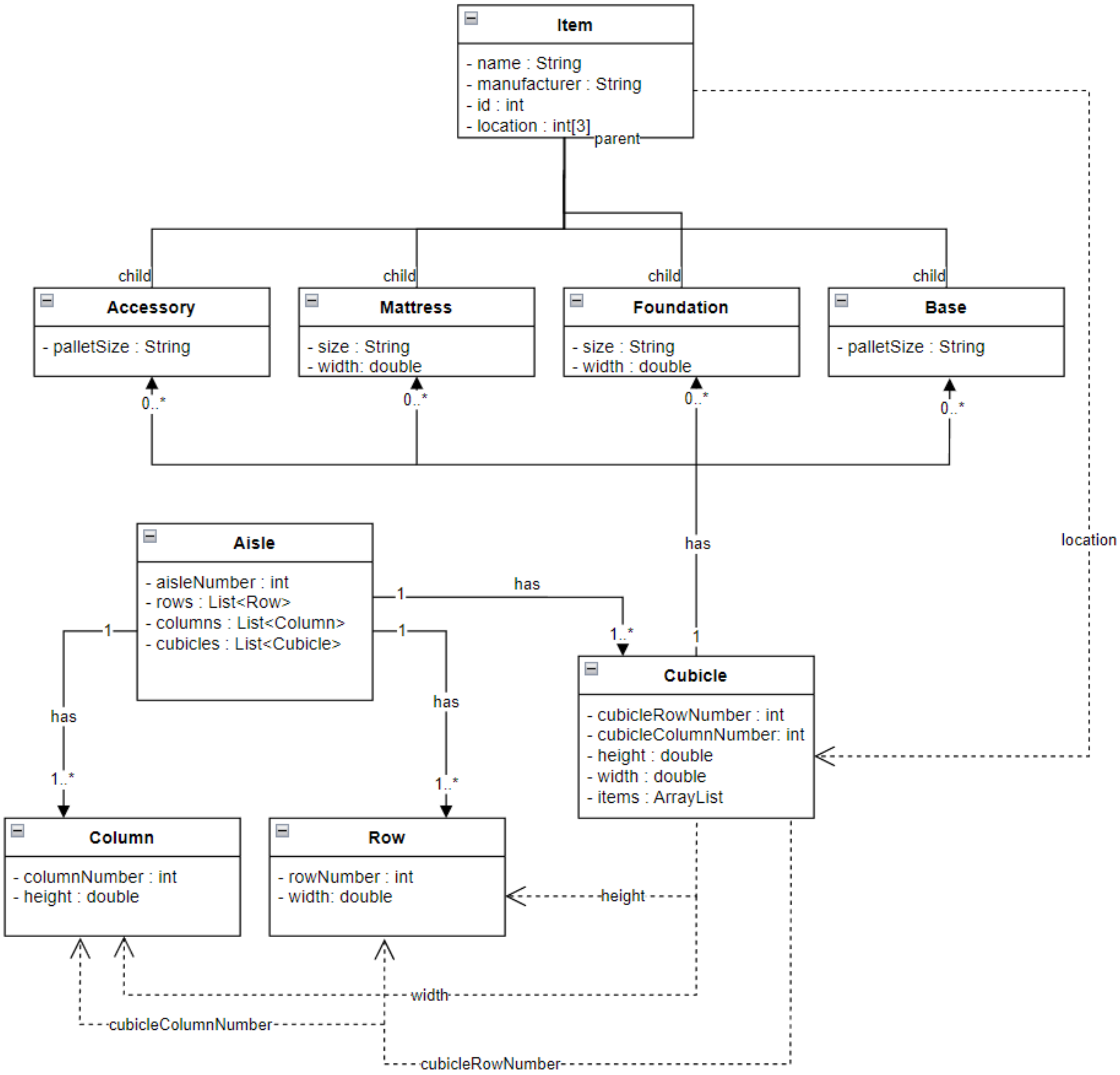
VI.2 Existing item classifications and attributes may not be enough for certain miscellaneous items. Section *VI.2 of Potential Requirements* could fix this limitation.

Detailed Design

Framework and Software Development Environment

Source code will be developed using Visual Studio 2019 compiler using a Windows Form template for GUI objects under the .NET 5.0 framework. Unit testing will also be done using Visual Studio 2019. Development process will consist of working on individual functionalities along with their respective GUI object elements while testing frequently to carefully document and trace where and when problems in source code occur.

Top Level Design Hierarchy



Class Methods and Processes

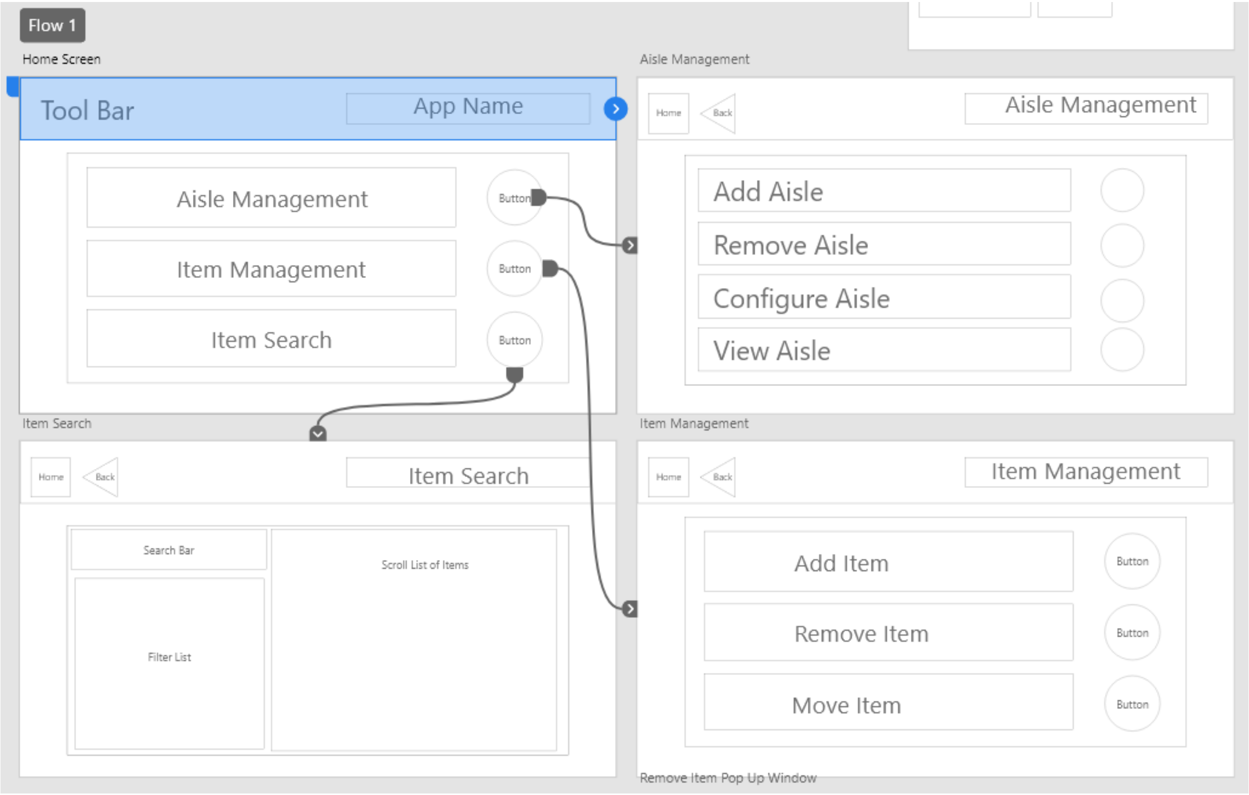
Non graphical-interface classes will contain getter and setter methods that will be used by user graphical interface objects from the Windows Form library.

Graphical interface objects such as buttons, lists, and checkboxes will be the primary way in which data and objects are manipulated in the program. For example, a button may create or remove an object.

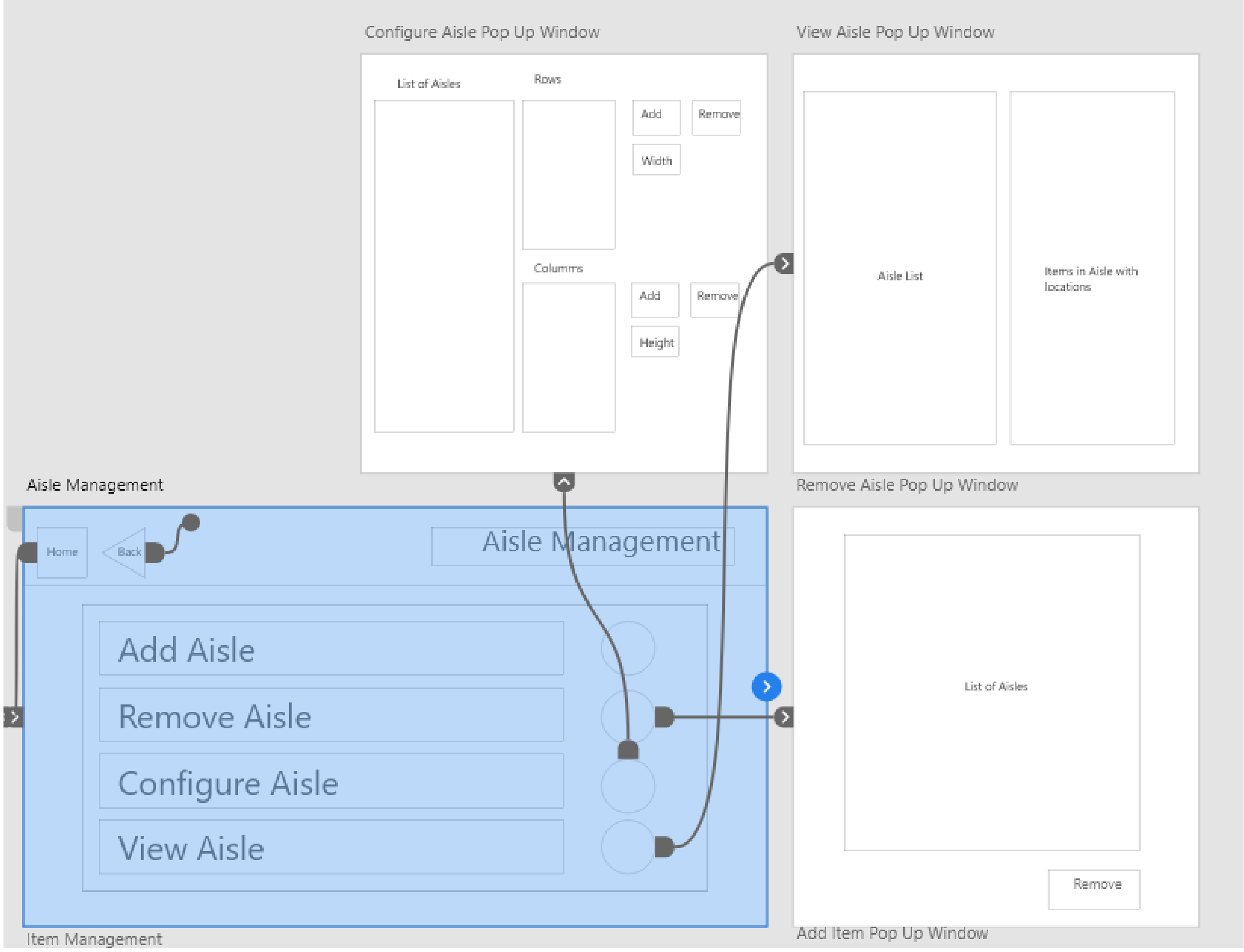
Some labels such as “Add Aisle” or “Move Item” describe their associated methods. Tying graphical interface objects to the manipulating methods of the Item, Aisle, and Cubicle class objects will provide a more user friendly experience, and a ease of use.

GUI Prototype – Wireframe

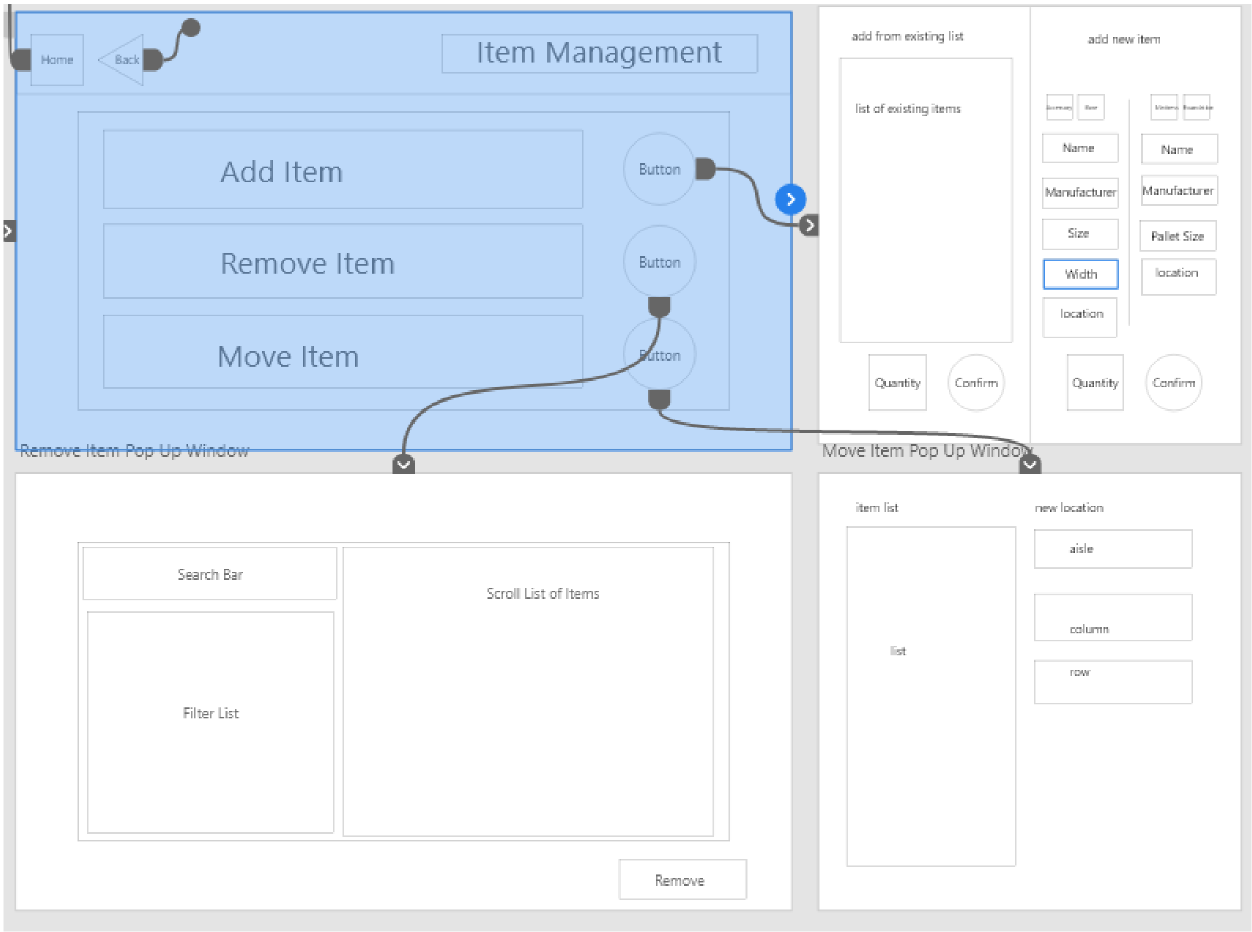
Home Screen:



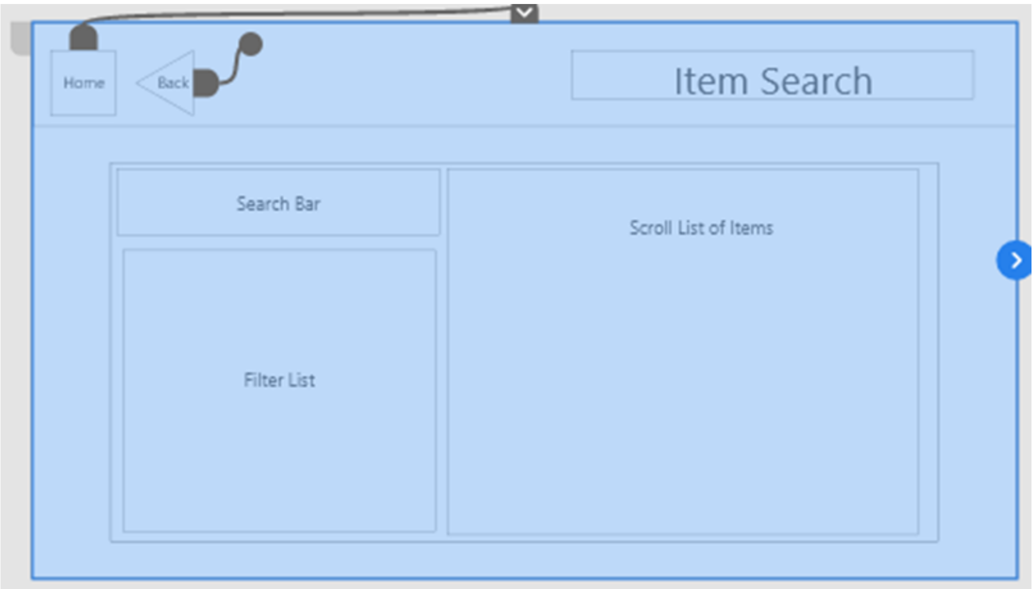
Aisle Management:



Item Management:



Item Search:



User Guide

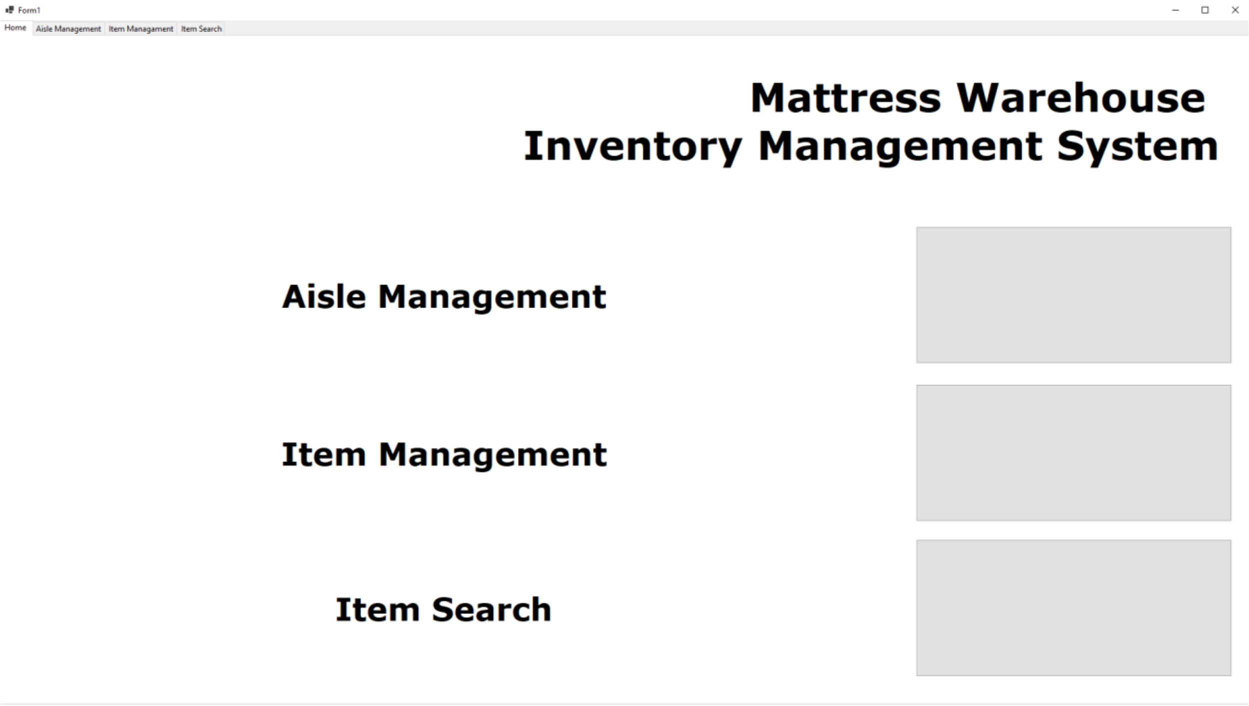
I. Introduction

The Mattress Warehouse Inventory Management System (MWIMS) creates a virtual Warehouse environment system that can be customized with a number of Aisles and Rows to store items created by the user. The software uses the following structure to store items. A Warehouse has a list of Aisles. An Aisle has a list of Rows, which can also be considered “floors” or “levels”. And Rows store a list of Items. A functioning warehouse has at least one Aisle and one Row to store items in.

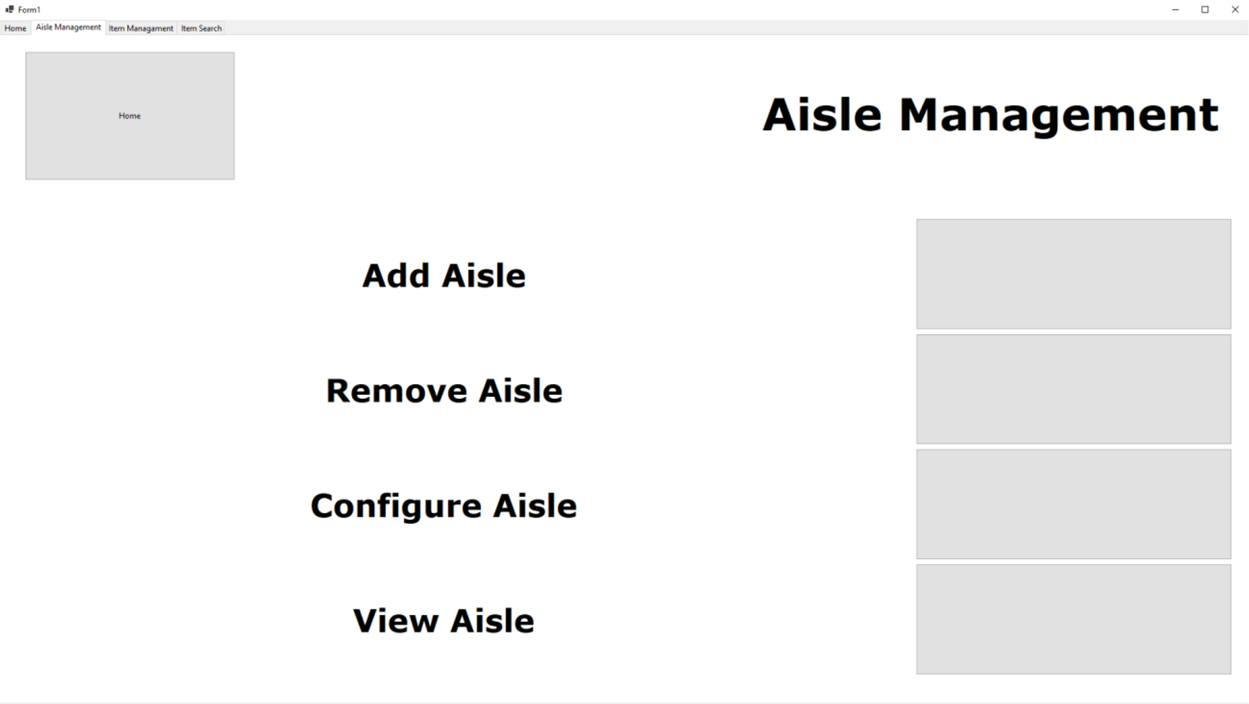
This guide will first explain every component of the software, and afterwards provide a first time use scenario on what to do when using the application for the first time.

II. Home Form

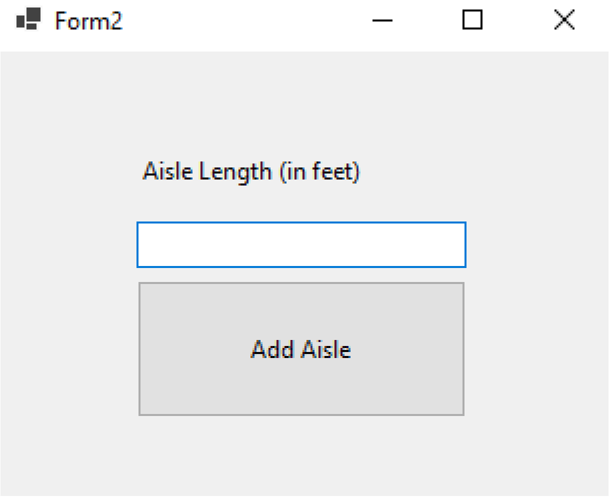
The MWIMS Home Form presents the user with the following three buttons: Aisle Management button, Item Management button, and Item Search button. Each button leads to a respective page from the Home Form and presents the user with additional choices. Each form tab also has a “Home” button on the top left of the page, which will take the user back to the home tab when pressed.



III. Aisle Management Tab

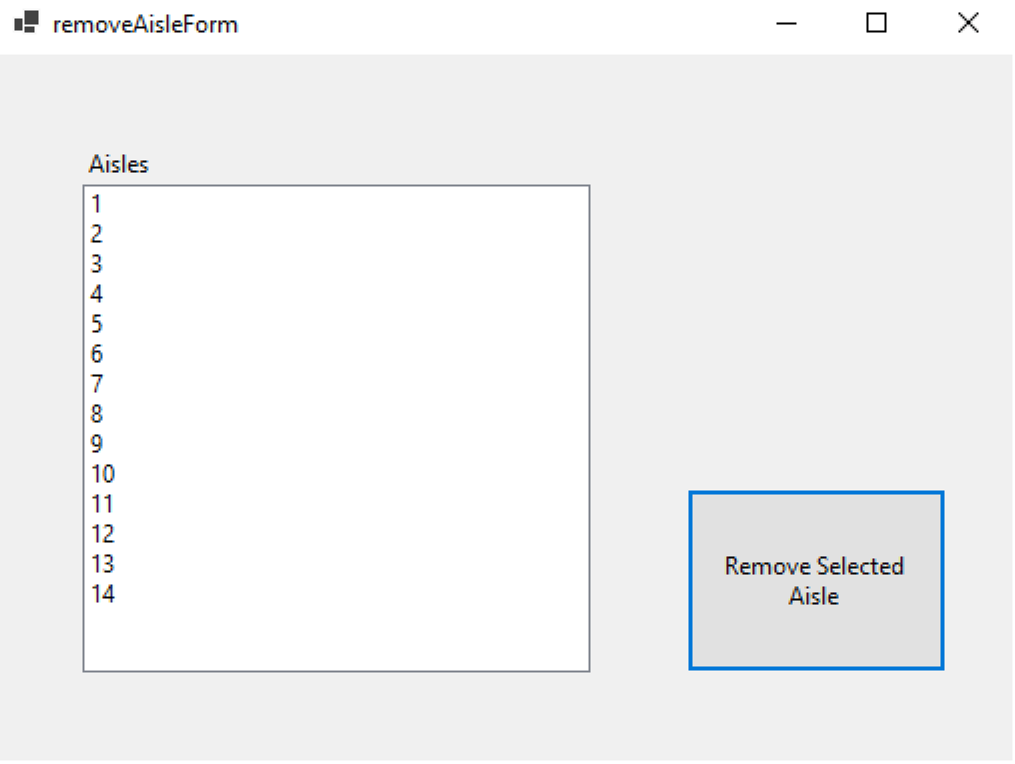


III.1 Add Aisle Form Button – Pressing this button will present the Add Aisle Form, a pop-up window with one input text box, and a button.



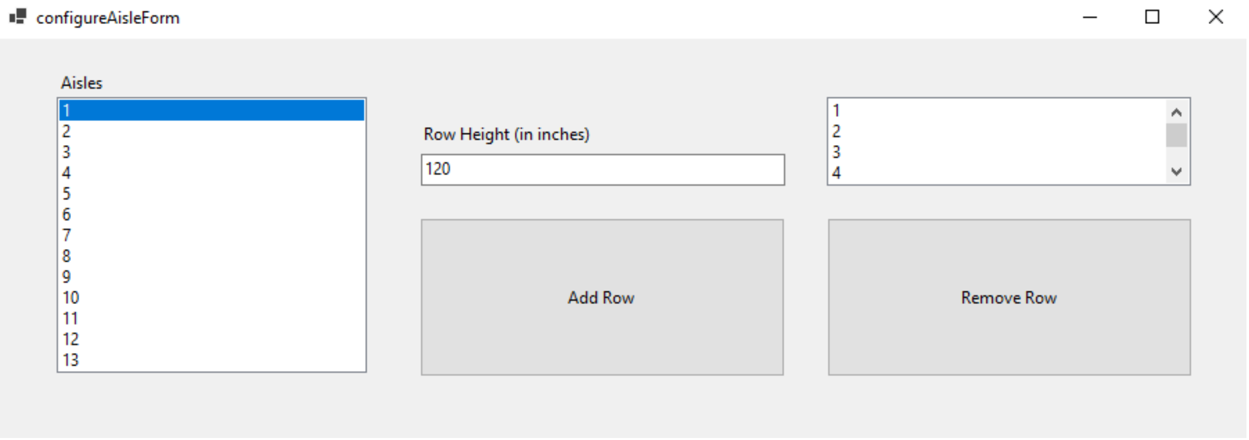
To add an Aisle, insert a numeric whole number for the aisle length in feet, and then press the “Add Aisle” button. Upon the button press, and aisle with the specified length will be created.

III.2 Remove Aisle Form Button – Pressing this button will present the Remove Aisle Form. This form is a pop-up window with a list and a button.



To remove an Aisle, select an aisle from the list on the left and press the "Remove Selected Aisle" button. The selected Aisle will be removed and deleted.

III.3 Configure Aisle Form Button – Pressing this button will present the configure form pop-up window. This window consists of a list of aisles, and input textbox, two buttons, and a list of rows.

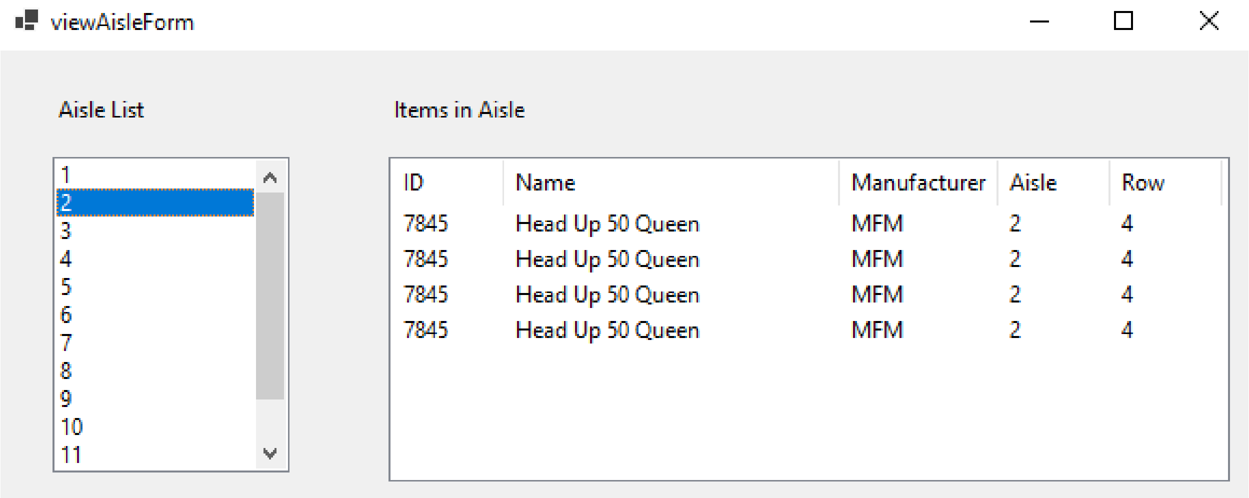


Row Height (in inches)
120

The configure aisle form window allows a user to add or remove Rows from an Aisle. To add a Row, select an Aisle from the Aisles list and enter a numeric input in the Row Height textbox. Pressing the "Add Row" button will create a row with the specified height in inches in the selected Aisle. To remove a

Row, select an Aisle, and then select the Row you'd like to remove. Pressing the “Remove Row” button will remove the selected Row from the selected Aisle.

III.4 View Aisle Form Button – Pressing this button present the View Aisle form pop-up window. This window consists of a list of Aisles and a list of Items.

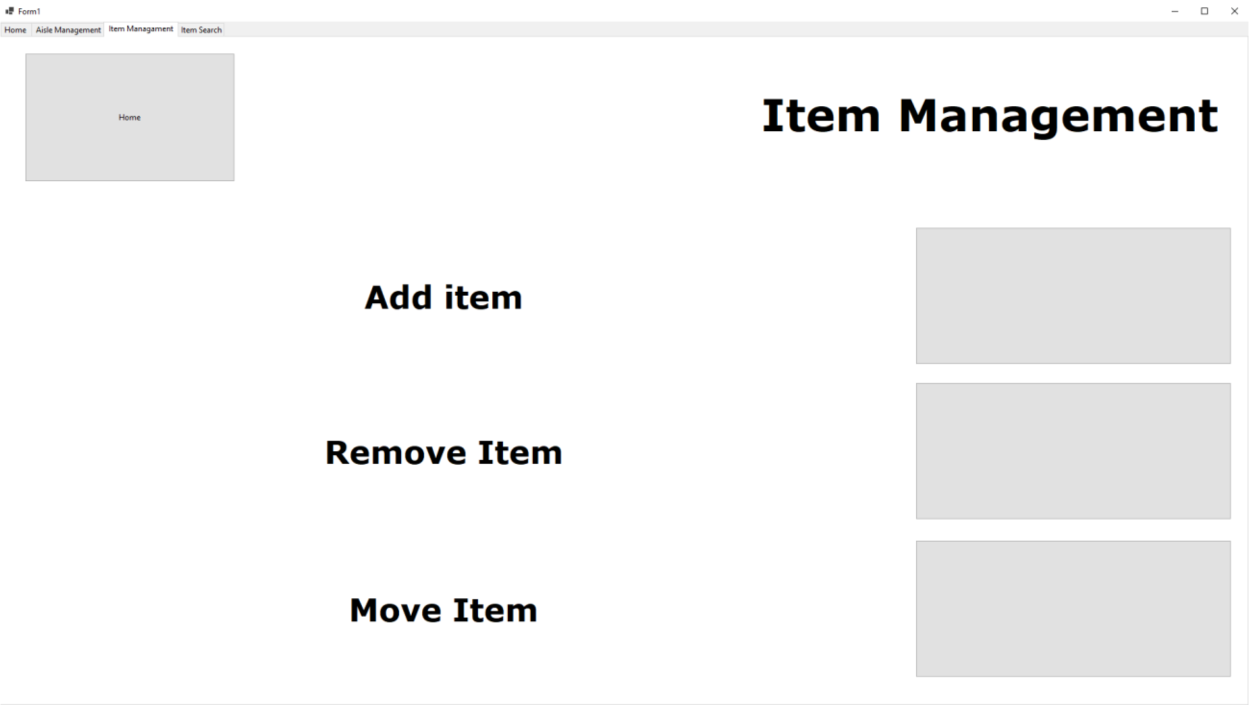


1
2
3
4
5
6
7
8
9
10
11

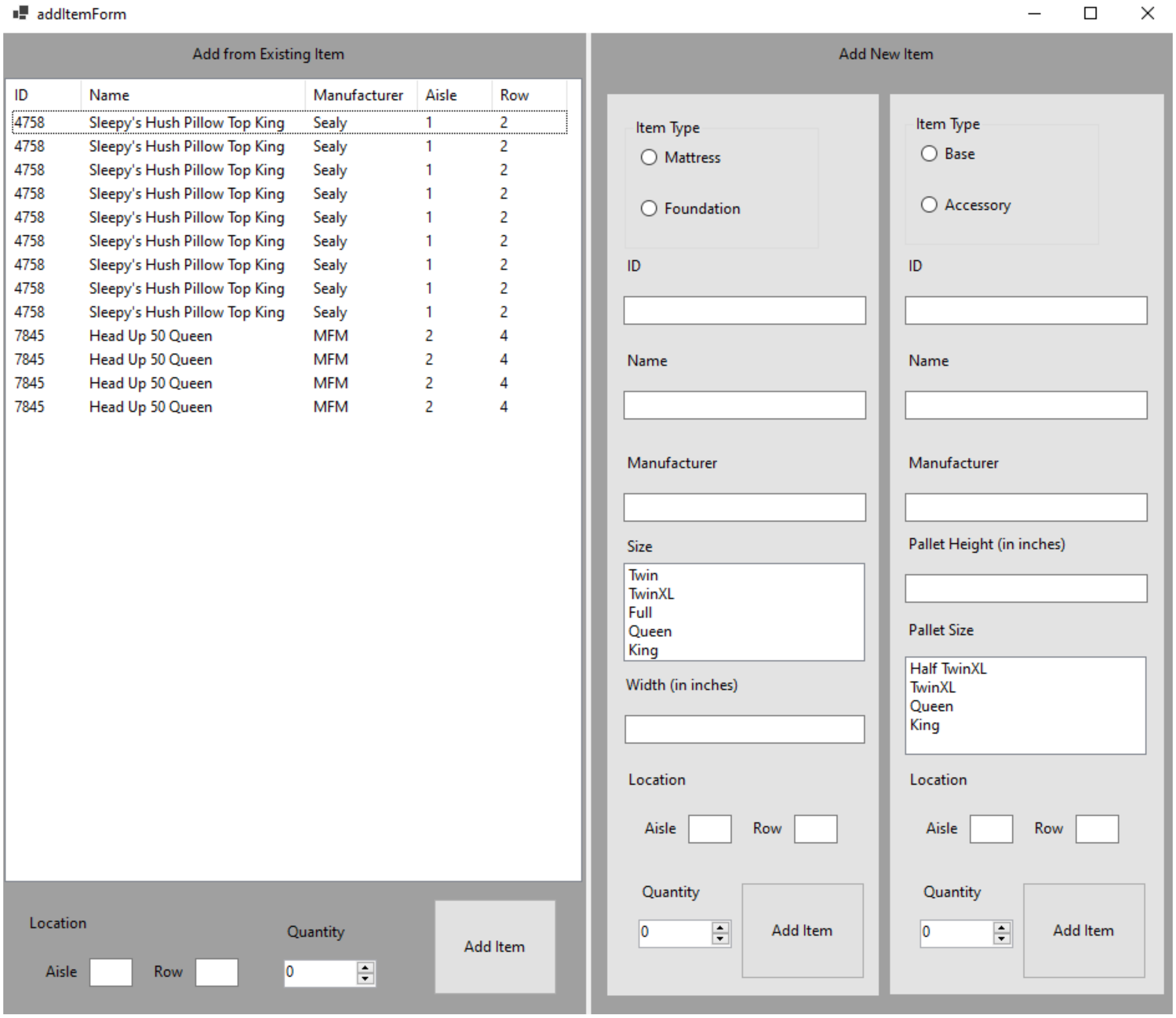
ID	Name	Manufacturer	Aisle	Row
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4

Selecting an Aisle from the list on the left will display all items in that Aisle.

IV. Item Management Tab



IV.1 Add Item Form Button – Pressing this button presents the Add Item Form pop-window.

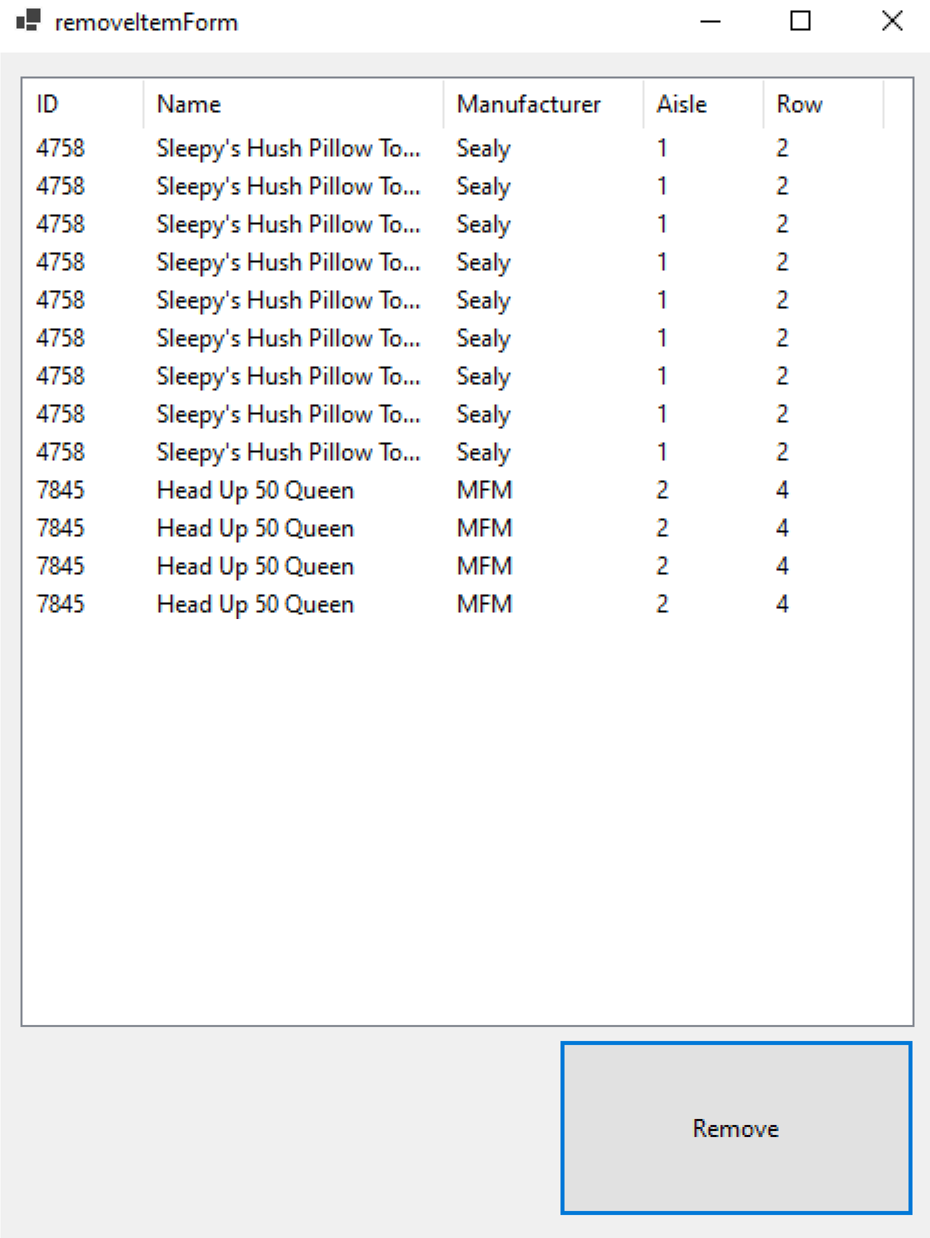


ID	Name	Manufacturer	Aisle	Row
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
4758	Sleepy's Hush Pillow Top King	Sealy	1	2
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4

There are two options to add an item. The first option is to select an item from the list on the left, input the number of an Aisle and number of a Row, select a quantity amount, and press the “Add Item” button. This will create one or more items exactly as the item that was selected, saving the user time from manually entering all item attributes to add more quantities of an item that already exists.

On the right side of the form window, a new item can be created. This side of the window is split into two segments, one for Mattress and Foundation type Items, and the other for Base and Accessory type Items. To create an item of either type, select one of the radio buttons under “Item Type”, then fill out the input text boxes with the relevant information, including to select a “Size” or Pallet Size” from the lists. Similar to creating an item from the existing items list, input an Aisle and Row number, select a quantity number, and click the respective “Add Item” button to add one or more items.

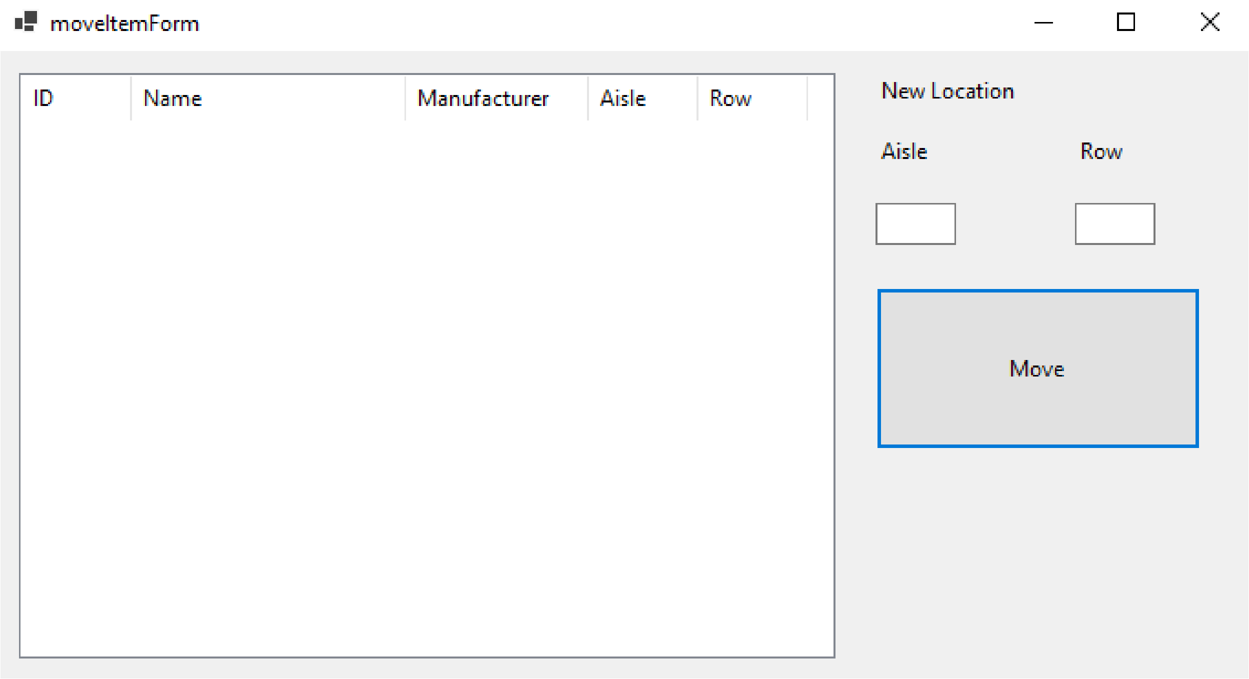
IV.2 Remove Item Form Button – Pressing this button will present the Remove Item form pop-up window. This form consists of a list of items, and a button.



ID	Name	Manufacturer	Aisle	Row
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
4758	Sleepy's Hush Pillow To...	Sealy	1	2
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4
7845	Head Up 50 Queen	MFM	2	4

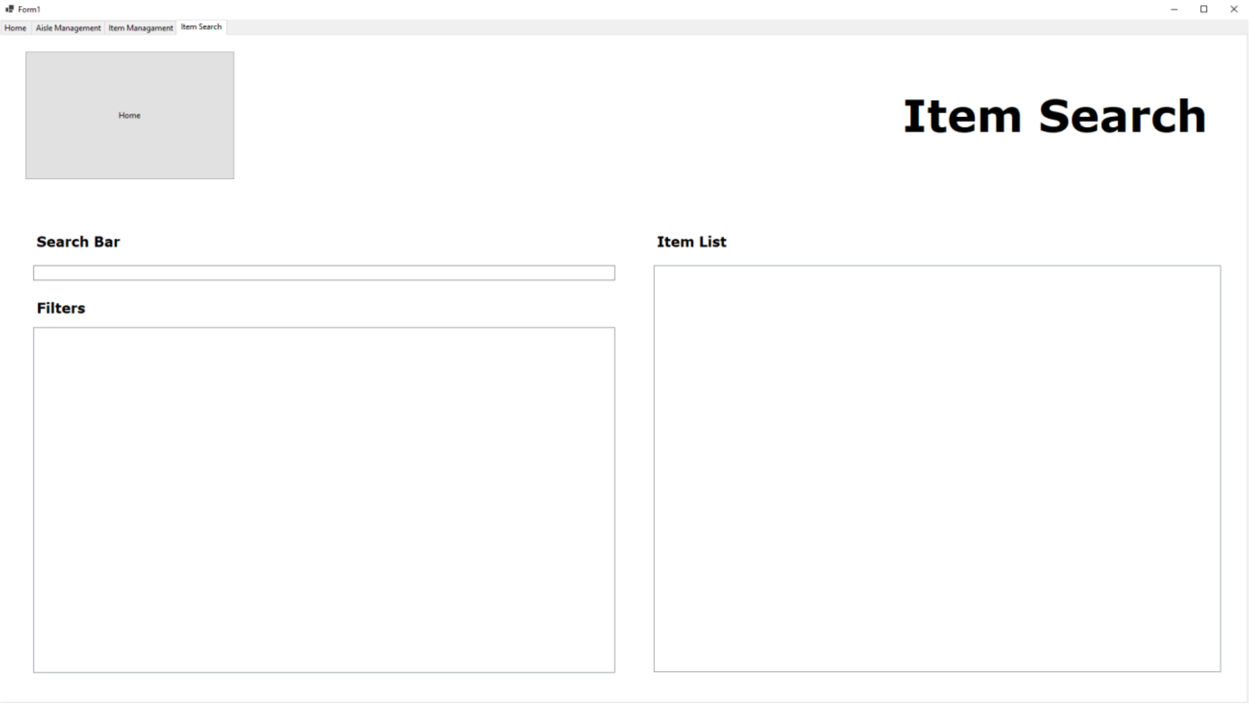
To remove an item, select an item from the list above and then click the ”Remove” button.

IV.3 Move Item Form Button – Pressing this button will present the Move Item form pop-up window. This window consists of a list of items, two input text boxes, and one button.



To move an item, first select the item from the list of items on the left. On the right, insert a new Aisle and Row number and then press the "Move" button.

V. Item Search Tab



VI. First Time User Guide

The first time the application is used, you should create at least one Aisle and one Row for that Aisle before creating any Items. First, press the Aisle Configuration button and then click Add Aisle. Create and Aisle and then close the window. Proceed to press the Aisle Configuration button and create one Row for your Aisle. Now you are free to create and place items on the first Row of the first Aisle.

VII. Software Limitations

Due to time constraints and limited expertise during development, the MWIMS has a few features lacking or limited.

VI.1 Data loading – Currently, all data such as created Aisles, Rows, and Items are serialized and saved under the directory MWIMS_Capstone\bin\Debug\net5.0-windows\warehouse.xml. While data is able to be saved, it is unable to be loaded and reused in the program's current state.

VI.2 Error handling – If the program user does something unexpectedly, the program currently does not have any safety measures or error handling measurements in place to prevent the abrupt crash of the program. For example, if a numeric whole number is expected to be inputted but a word is instead used, the program will not handle this error correctly and crash.

VI.3 Currently, an item may with a height greater than a row height be added, and the combined with of items may be greater than the length of the aisle they are in. This is not an intended behavior of the program.

VI.4 The Item Search feature in the Item Search Form Tab has not been implemented.

Conclusions

The final product submitted did not meet all of the *project proposal* and *detailed design* elements, but a substantial number of features were implemented. The product is a proof of concept that the system can be implemented and function properly. Expected outputs are present, and features, when properly used, work as intended. With more development time, all of the originally proposed features could be fully implemented along with safety measurements to prevent the program from stopping due to unexpected values.

References and Resources

W3 Schools C# Tutorial – Utilized to give myself a refresher on C# syntax and C# object-oriented programming concepts.

Stack Overflow – Utilized to access problems involving C# object-oriented programming, and Windows Form questions.

C# Discord Server – A community utilized to request help when an answer could not be found on the internet.

Microsoft .NET Documentation – Utilized for reference on C# syntax, object-oriented programming concepts, and Windows Form documentation.

Geeks for Geeks - Utilized for reference on C# syntax, object-oriented programming concepts, and Windows Form documentation.