Criterion B

Input Output Tables

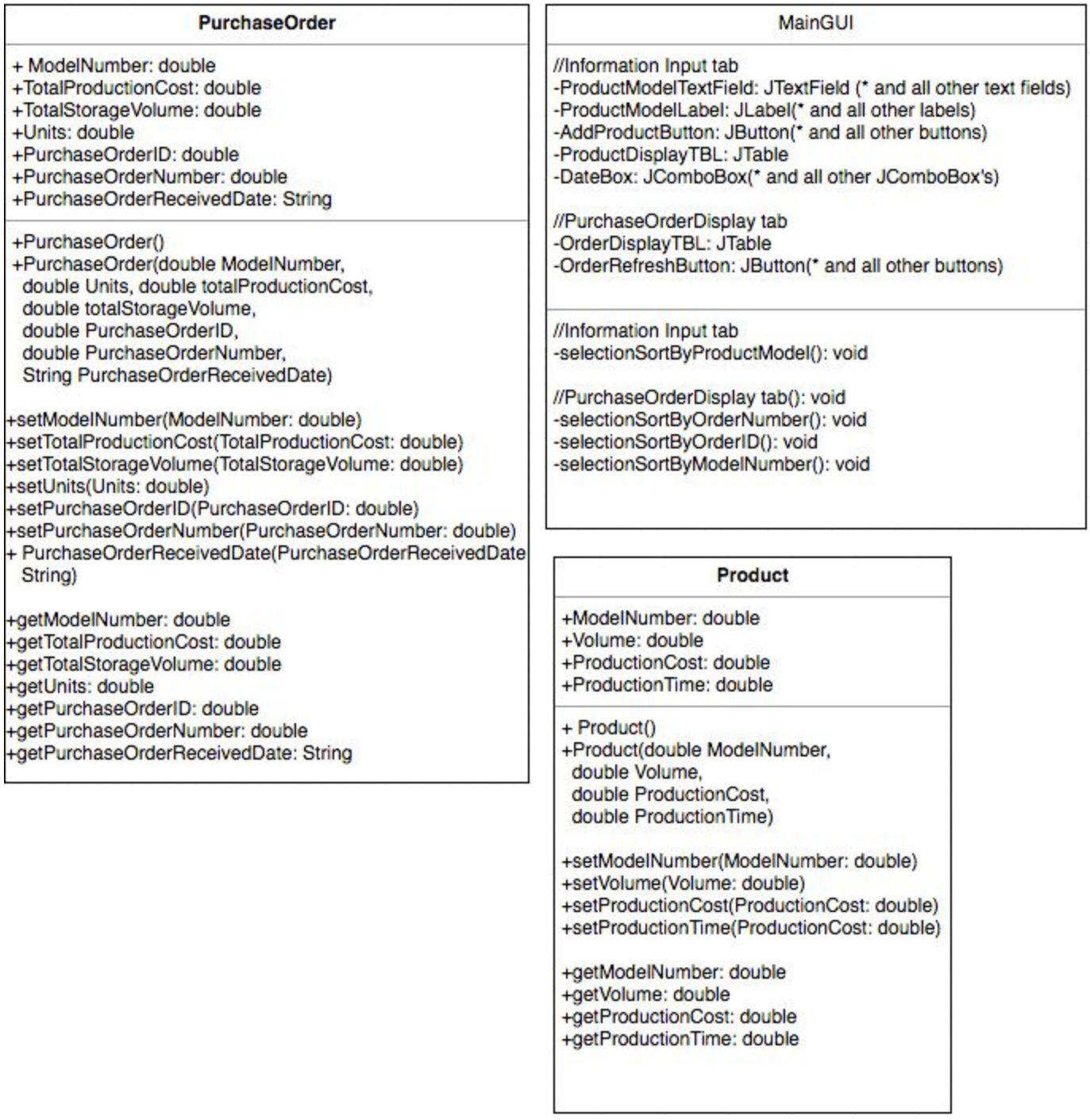
Inputs:

Inputs	Data Type	Examples
Model Number	double	2456
Volume	double	8
Production Cost	double	15
Production Time	double	10
Units	double	5
Purchase Order ID	double	2118
Purchase Order Number	double	2157
Month	int	11
Day	int	09
Year	int	2019

Outputs:

Outputs	Data Type	Examples
Total Storage Volume	double	1268
Total Production Cost	double	5684
Purchase Order Received Date	String	11/09/2019

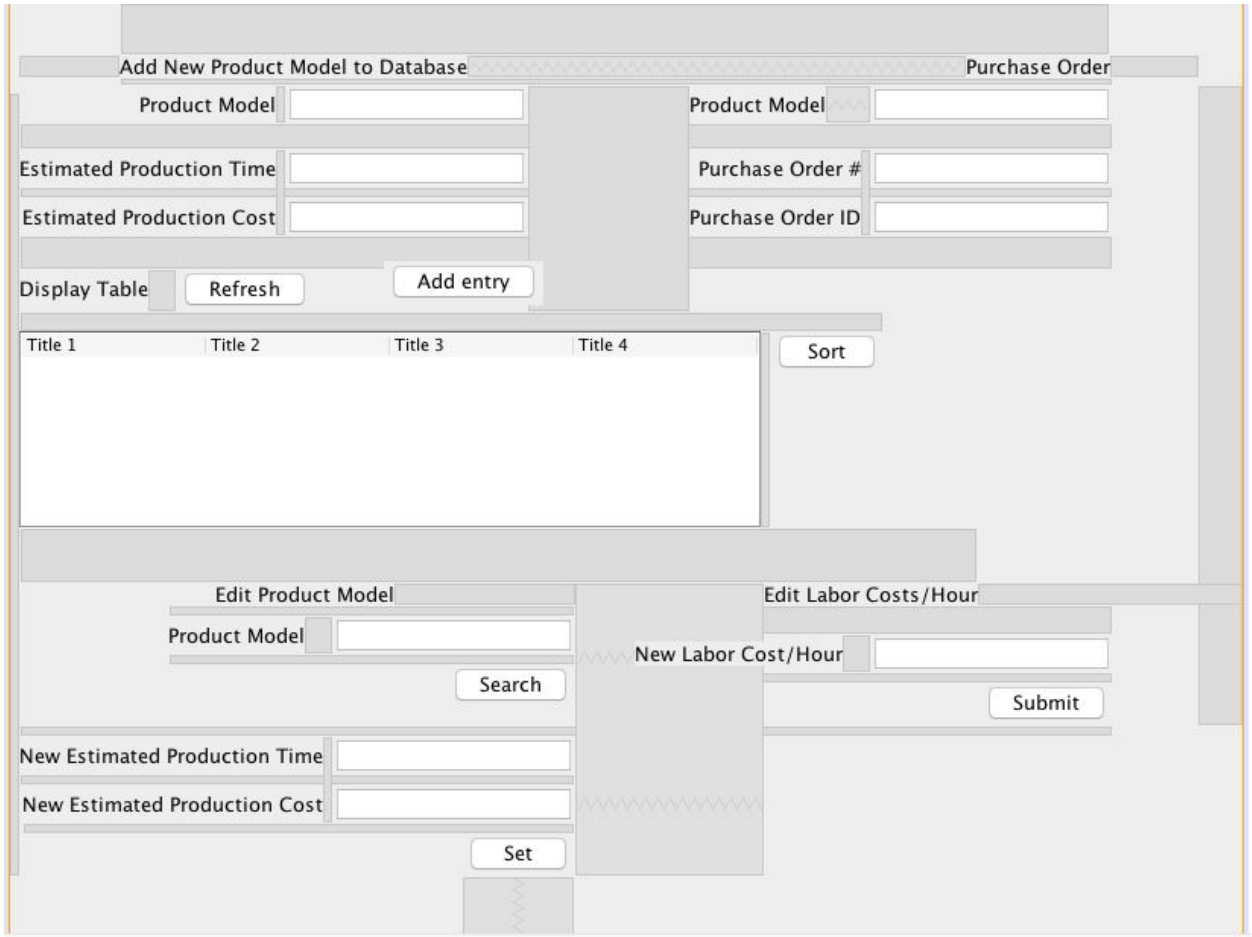
Class Diagrams



Prototyping Process

Initial Prototype Screenshots

jTextField1	jTextField2	jTextField3	jTextField4	jTextField5	jTextField6	jTextField7
jTextField8	jTextField9	jTextField10	jTextField11	jTextField12	jTextField13	jTextField14
jTextField15	jTextField16	jTextField17	jTextField18	jTextField19	jTextField20	jTextField21
jTextField22	jTextField23	jTextField24	jTextField25	jTextField26	jTextField27	jTextField28
jTextField29	jTextField30	jTextField31	jTextField32	jTextField33	jTextField34	jTextField35

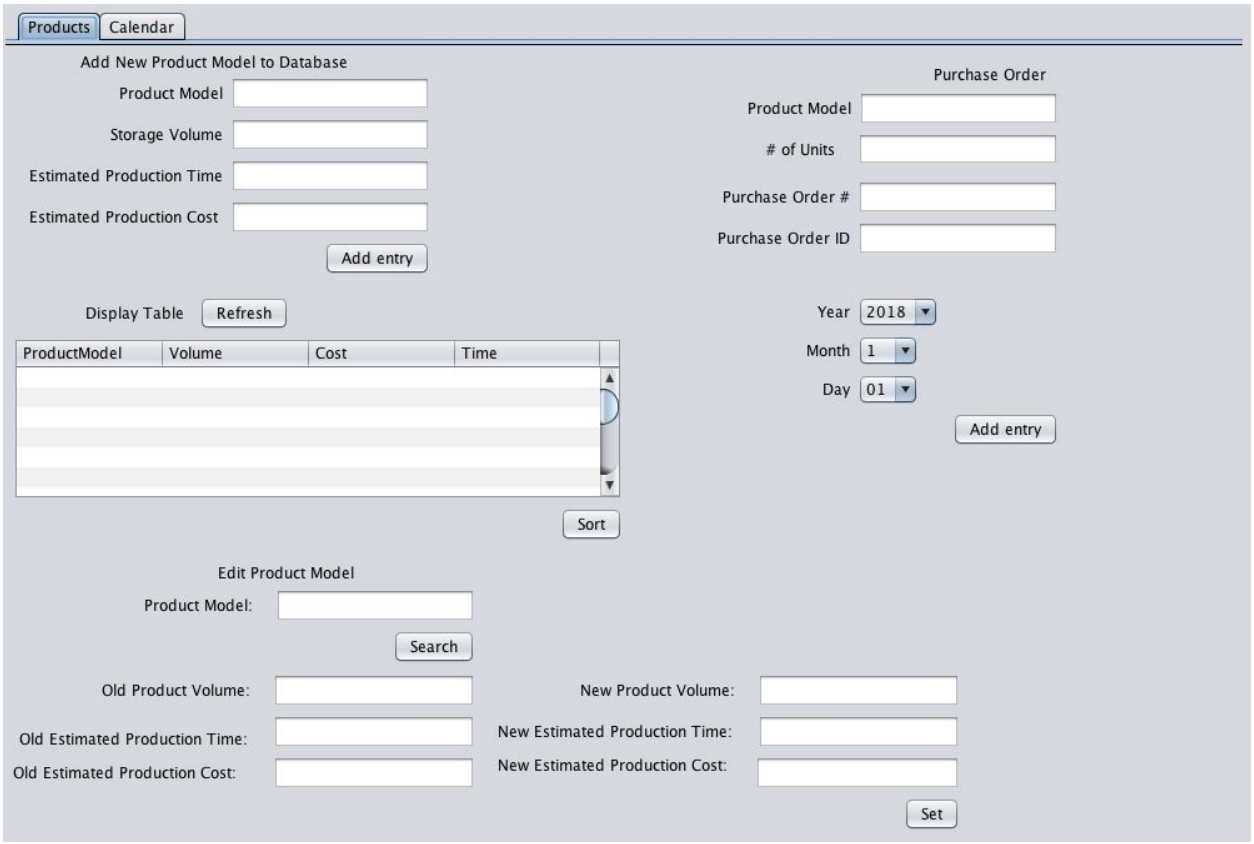


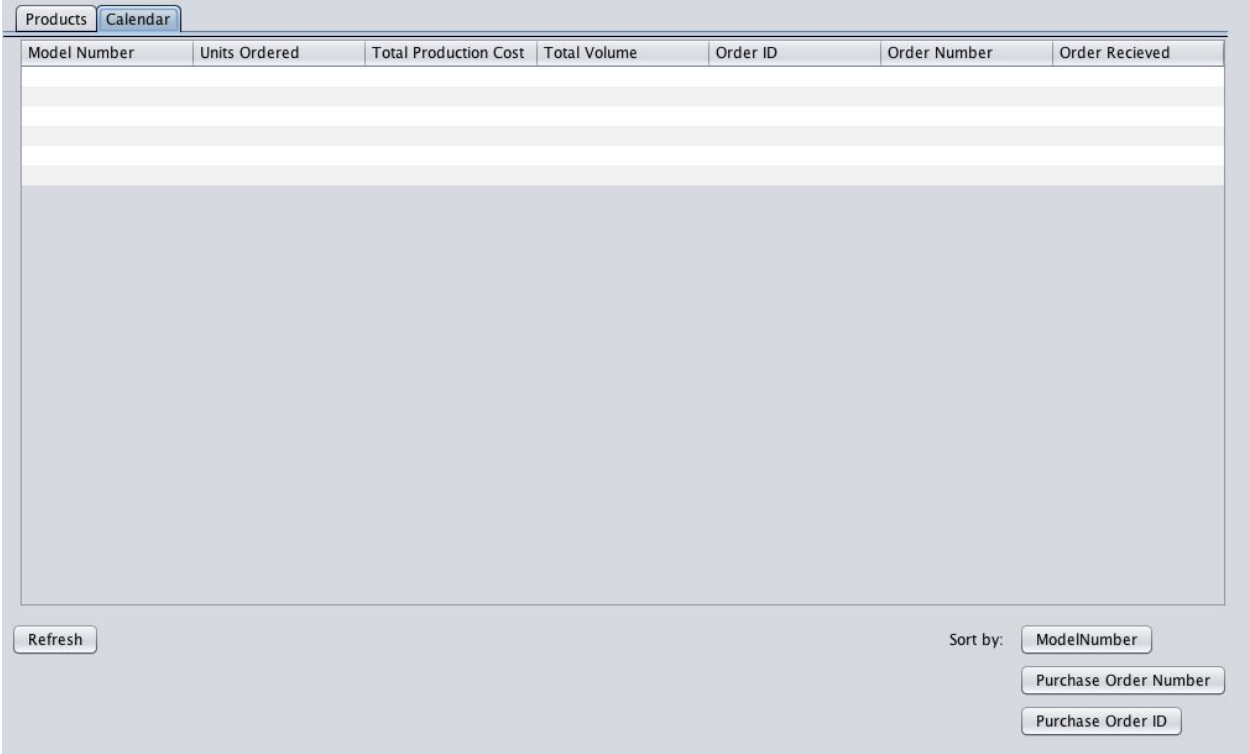
Changes Proposed

After interviewing with the client, he suggested these things for the program.

- suggested the removal of the labor cost function as the firm calculates the cost for that separately from the production cost of the purchase order.
- suggested that the way that information on purchase orders are displayed may not necessarily have to be a calendar as it may be a harder to comprehend for users.
- Suggested more JTextFields to display information about the product when editing each product model
- Suggested a way to store when the purchase order was received.

Final Prototype





Chronological Development Plan

1. Create Initial GUI prototype

2. Create all necessary Database structures
    - Product array that stores ModelNumber, Volume, ProductionTime, and ProductionCost
    - PurchaseOrder array that stores ModelNumber, Units, TotalProductionCost, TotalStorageVolume, PurchaseOrderID, PurchaseOrderNumber, and PurchaseOrderReceivedDate

3. Create formula that calculates total production cost and time
    - Create code that draws on product array to get values for Volume and ProductionCost of each item in Product Array
    - Create algorithm that calculates TotalProductionTime and TotalProductionCost based on units ordered of each product model
4. Create code to search for different elements within an array
    - Code that searches for a product model number and displays their corresponding production costs and times
    - Different elements should be able to be organized into numerical order by either production cost or time.

5. Further refine and improve GUI

6. Create algorithm that takes user input and stores them into Array

7. Create display table
    - Displays information on each product model
    - Displays information on each purchase order

8. Create Sorting Algorithms
    - Sorting algorithm by product model
    - Sorting algorithm by purchase order ID
    - Sorting algorithm by purchase order number
    - Sorting algorithm by model number

9. Finalize everything and fix any remaining issues.

Record of Tasks

Task Number	Planned Action	Planned outcome	Time estimated	Target completion date	Criterion
1	Creating success criteria	finishing the success criteria and showing it to client	30 minutes	October 16	A
2	Create GUI prototype	GUI prototype finished	30 minutes	October 18	B
3	Finish Criterion A	Criterion A finished	2 hours	October 19	A
4	Showing GUI prototype to client	GUI prototype showed to client and feedback received	40 minutes	October 20	B/P

5	Improving GUI prototype based on feedback	Finished implementing all suggestions of client into GUI	2 hours	October 30	B/P
6	Creating Arrays and all other data structures needed	Data structures completed	3 hours	November 4	P
7	Naming all the variables and the GUI	finish with all elements in the GUI appropriately named	90 minutes	November 6	P
8	Code the sorting and searching function of the array	get it partially working	90 minutes	November 10	P
9	Code the sorting and searching function of the array	Get it fully working	3 hours	November 13	P
10	Code the code that will allow client adding product models and purchase orders into the array	get some of it to work	90 minutes	November 14	P
11	Code the code that will allow client	get it nearly completed	90 minutes	November 15	P

	adding product models and purchase orders into the array				
12	Code the code that will allow client adding product models and purchase orders into the array	get the code completed	90 minutes	November 16	P
13	Start Work on Criterion C	Get it partially done	2 hours	January 6	C
14	Work on Criterion C	Get it nearly done	2 hours	January 11	C
15	Work on Criterion C	Finish Criterion C	2 hours	January 14	C
16	Further refine code	GUI improved, some glitches and issues fixed	4 hours	February 10	P
17	Finishing up Criterion B	Finish Criterion B	3 Hours	March 23	B
18	Show code to Client for final feedback	Get feedback	30 minutes	March 23	E
19	Finalize Code	Finish all code and fixed all issues with code	5 hours	March 25	P

20	Do Criterion D Video	Video Recorded	5 minutes	March 25	D
21	Evaluate finished code based on success criteria	Code evaluated	1 hour	March 26	E

Testing Plan

Things done on “Information Input” tab

- Product model number is input
- Storage volume, production cost and production time associated with said product model is also input
- Instances of Product class are made and assigned to an array of products
- Product model number , # of units, purchase order number, and purchase order ID associated with a purchase order are input
- Instances of purchaseOrder class are made and assigned to an array of purchaseOrders
- Product model of an existing product model in the array is input in “SearchProductModelTextField”
- New product volume, new production time, new production cost are input

Normal Functioning

- Product model such as 123 and a storage volume, production time, and production cost of 12, 34, and 56 are inputted respectively so that the product instance can be made
- Product model such as 123 and units, purchase order ID, and purchase order are input so that total production cost and total storage volume can be calculated, resulting in an instance of purchase order being made.
- When a product model is input into “SearchProductModelTextField” Storage volume, production cost, and production time associated with said product model is output in corresponding text fields
- New product volume, new production time, new production cost inputted replace the current ones in array

Abnormal

- If wrong data type is input into any fields such as a string “a12” instead of a double “123”, text fields will not clear, indicating an error.

Things done on “Purchase Order Display” tab

- Refresh and sort buttons can be pressed

- Order of items in table changes/updates according to which button is pressed

Normal Functioning

- When refresh is pressed, table updates new information that were input into purchaseOrder array
- When any of the sort button is pressed, it sorts data in table from least to greatest by 1 of three things: purchase order ID, purchase order number, and product model number

Abnormal

- Table doesn't update when refresh is pressed
- Table does not sort properly