Quarter Programming Project

<u>Logistics Application</u>

Overview

Logistics: The process of planning, implementing, and controlling the efficient, effective flow and storage of goods from the point of origin to a destination for the purpose of conforming to customer requirements.

Our application will process orders to move items (products) from one (or more) facilities to a destination facility.

These items must be processed at the source facility (retrieved from stock, packaged, packed into containers).

The containered items are then loaded onto transport trucks.

These trucks then transport the ordered items over approved shipping routes.

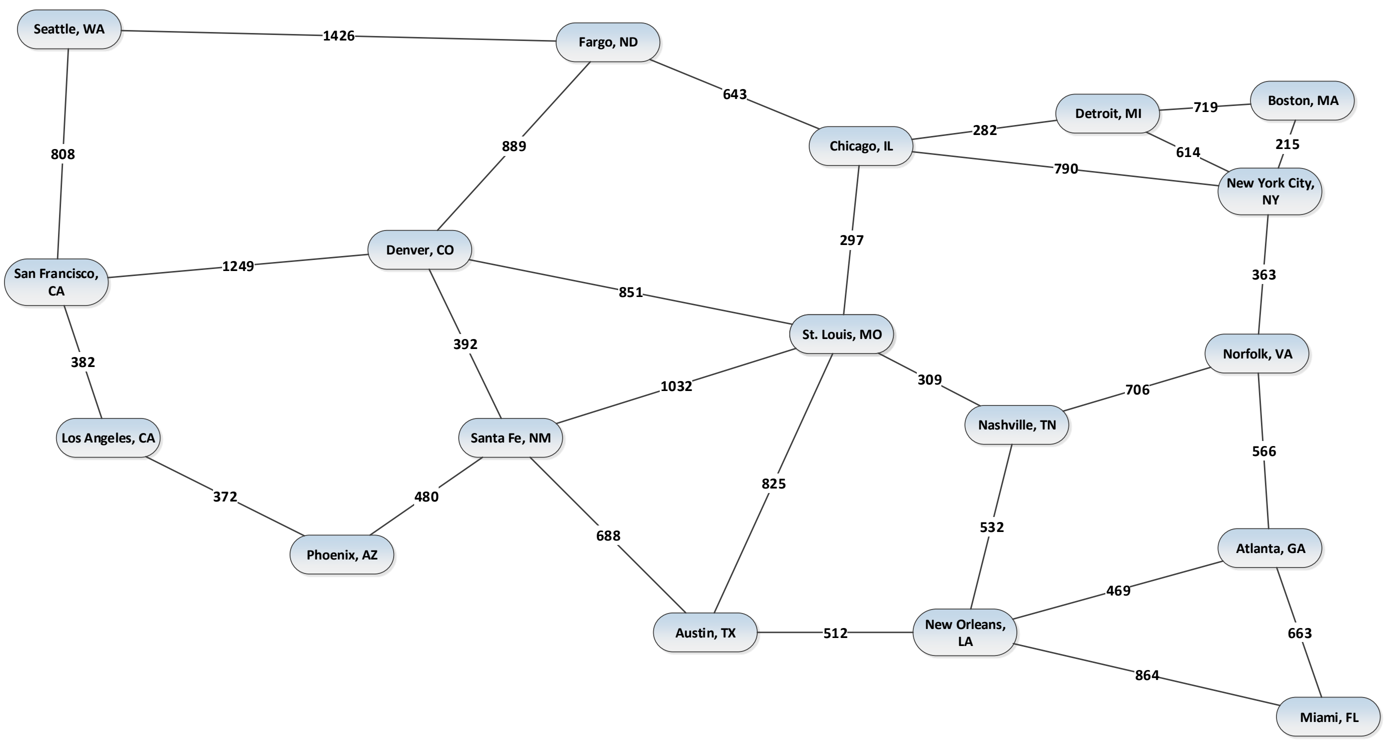
The ordered items are unloaded, unpacked and delivered to their destination facility.

Our project is to develop an application that will perform these Order Processing & Logistics functions – it will automatically process customer orders – a very time-intensive process when done manually. The input of information, order processing, and results generation should all be handled by your application. All required details can be found in this document.

Application Characteristics

Transportation

Our application will work with a network of “facilities”. Facilities can either act as an order destination (the delivery destination for the items in an order) or an item source (a facility from which the items are taken). Below is a “map” of the facilities (18 of them) that must be represented in your application. The available transportation links between facilities are represented by the lines between facilities. The distances between facilities (in miles) are shown on the lines. *Input details for facilities and the transportation network appear later in this document.*



Part of processing an order includes determining the best (i.e., shortest) path between facilities. For example, processing an order with destination “Denver” involves finding the shortest path to one of 3 facilities that have the desired item (assume they are “Chicago”, “Miami”, and “Norfolk”). While the network has many possible paths between Denver and the 3 listed facilities, the shortest paths would be:

- Chicago, IL to Denver, CO: [Chicago, IL - St. Louis, MO – Denver, CO] = 1,148 mi
- Miami, FL to Denver, CO: [Miami, FL - New Orleans, LA - Austin, TX - Santa Fe, NM - Denver, CO] = 2,456 mi
- Norfolk, VA to Denver, CO: [Norfolk, VA - Nashville, TN - St. Louis, MO - Denver, CO] = 1,866 mi

Your application must be capable of determining the shortest path between *any* 2 facilities. The distance between facilities should be represented as the travel time (in days):

Travel Time = Distance (miles) / (driving hours per day * average miles per hour)

(“driving hours per day” and “average miles per hour” should be input parameters to your application. Use the values 8 hours and 50 mph for these parameters).

For example (Chicago, IL to Denver, CO):

Travel Time = 1,148 mi / (8 hours per day * 50 mph) = 1,148 mi / 400 = **2.87 days**

Orders

An Order represents a request to have items (products) moved from one or more facilities to a destination facility. An Order should consist of the following information:

Order:		Example:	
• Order Time:	An "int" start day number	• Order Time:	Day 4
• Order Id:	A String	• Order Id:	ORD103
• Destination:	A String (a known destination)	• Destination:	Denver, CO
• List of "Order Items":		• List of "Order Items":	
○ Item ID: A String ,	Quantity: An int	○ Item ID: ABC123,	Quantity: 180
○ Item ID: A String ,	Quantity: An int	○ Item ID: ASD567,	Quantity: 50
○ …		○ …	

The "Order Time" field is the day on which the order was submitted (i.e., Day 1, Day 10, etc.)

Schedules

All facilities have limited processing capabilities. The processing refers to packing items and preparing them for shipment. Facilities can process a fixed number of items per day (each facility will have its own processing rate). Once the number of items for a day has been reached at a facility, the next available day must be used to continue processing. The amount of time it will take a facility to process a request is based upon the number of items requested, the facility's items-per-day rate, and the facility's available days.

For example, assume the "Seattle, WA" facility can process 10 items per day. Initially, the facility has no work booked (i.e., time already allocated to previous requests) so you can think of its schedule as follows:

Seattle, WA (Processing Rate: 10 Items per Day)													
Day #	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	…
Items that can be processed	10	10	10	10	10	10	10	10	10	10	10	10	…

If this facility was asked to process 26 items starting on Day 1, then the schedule would be booked starting at the first open day (Day 1) as follows: Day 1 (10 items), Day 2 (10 items), Day 3 (6 items) – totaling 26 items. As you can see, processing would take 3 days (rounding up to a full day). The resulting schedule after booking this time at the facility would look as follows:

Seattle, WA (Processing Rate: 10 Items per Day)													
Day #	***1***	***2***	***3***	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	…
Items that can be processed	**0**	**0**	**4**	10	10	10	10	10	10	10	10	10	…

If this facility was then asked to process 33 items for another order (also starting at Day 1), the schedule would be booked starting at the first *open* day (Day 3) as follows: Day 3 (4 items), Day 4 (10 items), Day 5 (10 items), Day 6 (9 items) – totaling 33 items. The resulting schedule after booking this time would look as follows:

Seattle, WA (Processing Rate: 10 Items per Day)													
Day #	*1*	*2*	***3***	***4***	***5***	***6***	*7*	*8*	*9*	*10*	*11*	*12*	…
Items that can be processed	0	0	**0**	**0**	**0**	**1**	10	10	10	10	10	10	…

<u>Order Processing</u>

The Order Processing procedure involves a variety of activities, and is shown below:

Assume an Order has the following 2 Order Items:

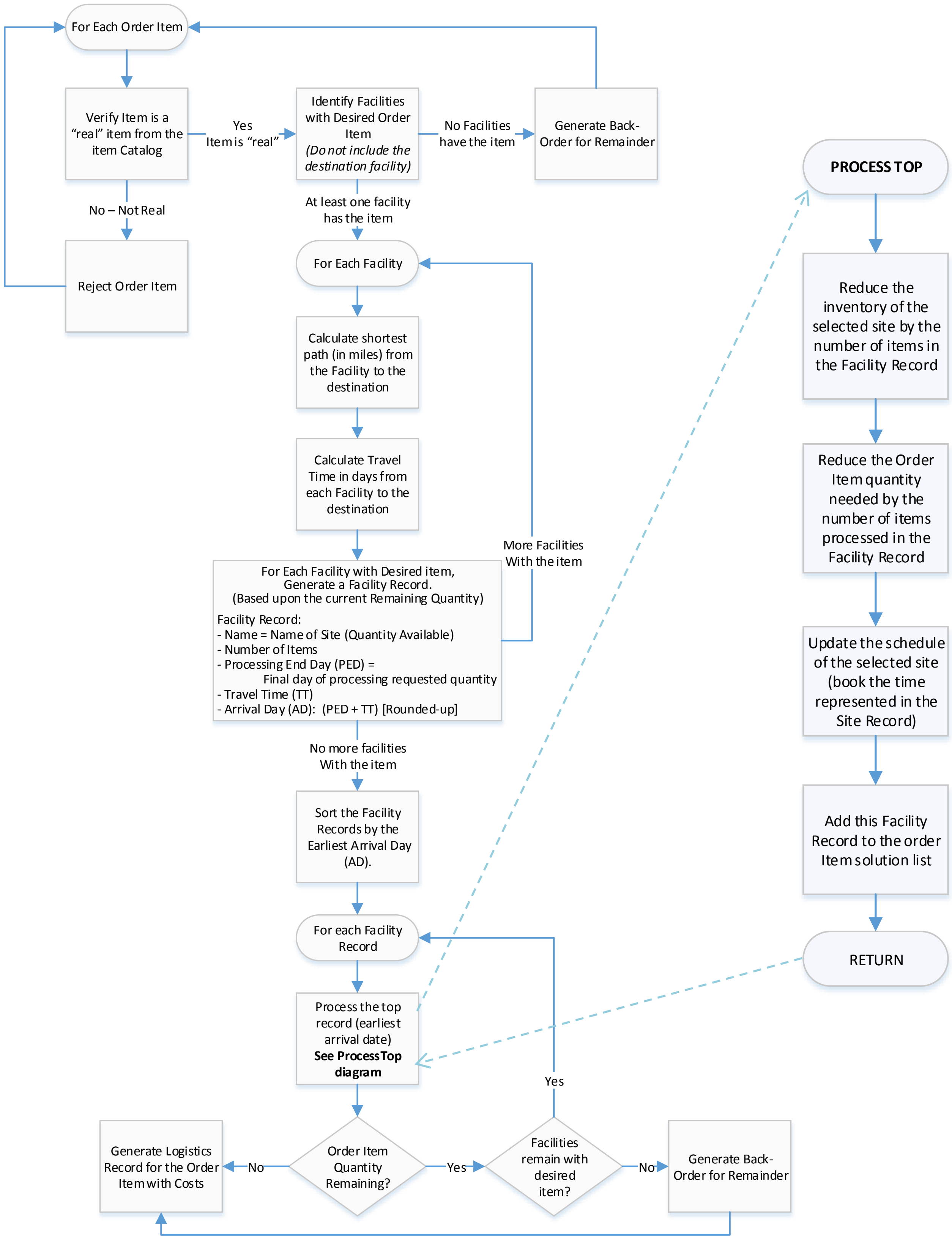
1. *Item ID: X499E,   Quantity 100*
2. *Item ID: XC670L,   Quantity 480*

For *each* of these 2 order items, do the following:

1) Identify all facilities with the desired item.
2) For each identified facility:
   a) Calculate the shortest path (in days) from the facility to the destination.
   b) Determine the days needed to process the items located at the facility.
   c) Add the travel time to the previously calculated processing end day to generate the "Arrival Day" of the item for that facility.
   d) Save this information as a Facility Record (*described later*) – a potential solution
3) Sort the records developed in step "2d" above by earliest (lowest) Arrival Day
4) Select the facility with the earliest (lowest) Arrival Day and do the following:
   a) Reduce the inventory of the item at that facility by the number of items taken
   b) Reduce the quantity of the item that is needed for the order by the amount taken from the selected facility
   c) Update the schedule of the selected site (book the days needed to process the items)
   d) Save this operation as part of your solution
5) If there is still more quantity of the item needed, go back to step 4 and continue the process. Otherwise proceed with step 6.
6) Compute the total cost of this item. An example appears later in this document (*see the "Order and Order Item Costs" section*).
7) Generate the complete logistics record for this order item (your set of solutions from step 4). An example appears later in this document (*see the "Order Item Logistics Record" section*).
8) If there are more items to process in this order, go back and repeat this process from step 1 with the next order items. Otherwise proceed with step 9.
9) Generate output. An example appears later in this document (see the "Order Output" section).

Order Item Processing

This workflow for processing *each* Order Item is detailed in the below diagrams:


For Each Order Item
Verify Item is a "real" item from the item Catalog
Yes Item is "real"
No – Not Real
Reject Order Item
Identify Facilities with Desired Order Item (Do not include the destination facility)
No Facilities have the item
Generate Back-Order for Remainder
At least one facility has the item
For Each Facility
Calculate shortest path (in miles) from the Facility to the destination
Calculate Travel Time in days from each Facility to the destination
More Facilities With the item
For Each Facility with Desired item, Generate a Facility Record. (Based upon the current Remaining Quantity)
Facility Record:
- Name = Name of Site (Quantity Available)
- Number of Items
- Processing End Day (PED) = Final day of processing requested quantity
- Travel Time (TT)
- Arrival Day (AD): (PED + TT) [Rounded-up]
No more facilities With the item
Sort the Facility Records by the Earliest Arrival Day (AD).
For each Facility Record
Process the top record (earliest arrival date) See Process Top diagram
Order Item Quantity Remaining?
No
Generate Logistics Record for the Order Item with Costs
Yes
Facilities remain with desired item?
Yes
No
Generate Back-Order for Remainder
PROCESS TOP
Reduce the inventory of the selected site by the number of items in the Facility Record
Reduce the Order Item quantity needed by the number of items processed in the Facility Record
Update the schedule of the selected site (book the time represented in the Site Record)
Add this Facility Record to the order Item solution list
RETURN

Order and Order Item Costs

The logistics cost for an Order in our application will be the sum of the logistics cost of the Order Items that make up the order. The logistics costs of an Order Item consist of:

(Item Cost + Facility Processing Cost + Transport Cost)

- The Item Cost is the cost of the item (an input value) multiplied by the quantity of the item.
- The Facility Processing Cost is the Facility's daily processing cost multiplied by the number of processing days. Each facility has its own daily processing cost. *(Partial days are pro-rated to the portion of the day used. For example, processing 7 of the 12 items a facility can process in a day would incur a cost of 58.3% -- 7 divided by 12 -- of the daily cost).*
- The Transport Cost is the Travel Time days (rounded UP to the next whole day) multiplied by the daily travel cost of $500 (this "$500" is constant cost throughout the application).

For example, examine the following logistics record for an item "ABC123". The costs are shown below:

Facility: Chicago, Items: 36, Processing End Day: 4, Travel Time: 1.98d, Arrival Day: 6, Cost: $12,520

- Item Cost = $290.00 (the item cost) * 36 (the quantity) = $10,440
- Facility Processing Cost = 3.6 days (3 full days, 1 day 60%) * $300/day (Chicago's processing cost) = $1,080
- Transport Cost = 2d (rounded up from 1.98) * $500 per day = $1,000
- **Total = ($10,440 + $2,550 + $1,000) = $12,520**

Order Item Logistics Record

An Order Item Logistics Record contains the "solution" plan to satisfy one item in an order. The record should contain the information necessary to develop the order's total cost. An Order Item Logistics Record should contain the following:

```
Item ID:  ABC123, Quantity: 180

Item Arrivals:
   •  Day 4: 36 (20.0%, 20.0% of total)
   •  Day 13: 61 (33.9%, 53.9% of total)
   •  Day 14: 28 (15.5%, 69.4% of total)
   •  Day 16: 55 (30.5%, 100% of total)

Logistics Details:
 1) Name: Chicago (36 of 180)               2) Name: Atlanta (61 of 180)
        Processing Start:  Day 1                   Processing Start:    Day 2
        Processing End:    Day 4                   Processing End:      Day 10
        Travel Start:      Day 5                   Travel Start:        Day 11
        Travel End:        Day 6                   Travel End:          Day 13


 3) Name: Dallas (55 of 180)                4) Name: Los Angeles (28 of 180)
        Processing Start:  Day 1                   Processing Start:    Day 2
        Processing End:    Day 11                  Processing End:      Day 6
        Travel Start:      Day 12                  Travel Start:        Day 7
        Travel End:        Day 16                  Travel End:          Day 14
```

Application Outputs

The output that should be generated by your application is dependent upon the orders it processes. For each order, the output should summarize the order and its processing solution. In addition to the order output, your application should also include a summary of the state of all facilities including its current inventory, a list of depleted items (items completely used up), and booking schedule.

Example Order Output

```
---------------------------------------------------------------------------------
Order #1
•  Order Id:      TO-001
•  Order Time:    Day 1
•  Destination:   Miami, FL
•  List of Order Items:
   o  Item ID:    ABC123,      Quantity: 180
   o  Item ID:    CR2032,      Quantity: 320

Processing Solution:
•  Total Cost:        $94,355
•  1st Delivery Day: 3
•  Last Delivery Day:  16
•  Order Items:
   Item ID   Quantity   Cost       # Sources Used   First Day    Last Day
   ABC123    180        $67,705    4                4            16
   CR2032    320        $26,650    1                3            6
---------------------------------------------------------------------------------
```

Facility Status Output

```
--------------------------------------------------------------------------------
Chicago, IL
-----------
Rate per Day: 10
Cost per Day: 300.0

Direct Links:
Detroit, MI (0.7d); Fargo, ND (1.6d); New York City, NY (2.0d); St. Louis, MO
(0.7d);

Active Inventory:
    Item ID      Quantity
    ABC123       60
    CT1928       20
    E241i        64
    RTF110       110
    XTP202       20

Depleted (Used-Up) Inventory: None

Schedule:
Day:            1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
Available:     10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10
--------------------------------------------------------------------------------
```

Input Data

The following data tables (Facilities & Network, Items, Facility Inventory, and Orders) should be used as input to your application. Each type of input should be located in its own file (preferably XML). Your application should load these files and store the data in whatever data structure you deem appropriate.

- Facilities & Network (18)

Facility [Rate/d @ Cost]	Transportation Links -- *Name (distance)*				
Seattle, WA [8/d @ $300]	*Fargo, ND (1426)*	*San Francisco, CA (808)*			
San Francisco, CA [12/d @ $300]	*Seattle, WA (808)*	*Denver, CO (1249)*	*Los Angeles, CA (382)*		
Los Angeles, CA [10/d @ $300]	*San Francisco, CA (382)*	*Phoenix, AZ (372)*			
Phoenix, AZ [8/d @ $300]	*Los Angeles, CA (372)*	*Santa Fe, NM (480)*			
Denver, CO [10/d @ $300]	*San Francisco, CA (1249)*	*Santa Fe, NM (392)*	*Fargo, ND (889)*	*St. Louis, MO (851)*	
Santa Fe, NM [8/d @ $300]	*Phoenix, AZ (480)*	*Denver, CO (392)*	*St. Louis, MO (1032)*	*Austin, TX (688)*	
Fargo, ND [8/d @ $300]	*Seattle, WA (1426)*	*Denver, CO (889)*	*Chicago, IL (643)*		
Austin, TX [10/d @ $300]	*Santa Fe, NM (688)*	*St. Louis, MO (825)*	*New Orleans, LA (512)*		
St. Louis, MO [12/d @ $300]	*Chicago, IL (297)*	*Nashville, TN (309)*	*Austin, TX (825)*	*Santa Fe, NM (1032)*	*Denver, CO (851)*
Chicago, IL [10/d @ $300]	*Fargo, ND (643)*	*St. Louis, MO (297)*	*New York City, NY (790)*	*Detroit, MI (282)*	
New Orleans, LA [10/d @ $300]	*Austin, TX (512)*	*Nashville, TN (532)*	*Atlanta, GA (469)*	*Miami, FL (864)*	
Nashville, TN [8/d @ $300]	*St. Louis, MO (309)*	*New Orleans, LA (532)*	*Norfolk, VA (706)*		
Detroit, MI [10/d @ $300]	*Chicago, IL (282)*	*New York City, NY (614)*	*Boston, MA (719)*		
Boston, MA [10/d @ $300]	*Detroit, MI (719)*	*New York City, NY (215)*			
New York City, NY [14/d @ $300]	*Boston, MA (215)*	*Detroit, MI (614)*	*Chicago, IL (790)*	*Norfolk, VA (363)*	
Norfolk, VA [10/d @ $300]	*New York City, NY (363)*	*Nashville, TN (706)*	*Atlanta, GA (566)*		
Atlanta, GA [10/d @ $300]	*Norfolk, VA (566)*	*New Orleans, LA (469)*	*Miami, FL (663)*		
Miami, FL [12/d @ $300]	*Atlanta, GA (663)*	*New Orleans, LA (864)*			

- Item Catalog (16)

Item Id (Price)					
ABC123 ($550)	CR2032 ($240)	CT1928 ($910)	E241i ($10,400)	JBL3100 ($180)	MM35P ($1,950)
PL132-C ($440)	PU238 ($2,200)	RL123A ($360)	RTF110 ($715)	RX100-3 ($642)	SN-241-L ($620)
SR71-D ($1,600)	XLK200B ($820)	XTP202 ($345)	ZTF109 ($1,100)		

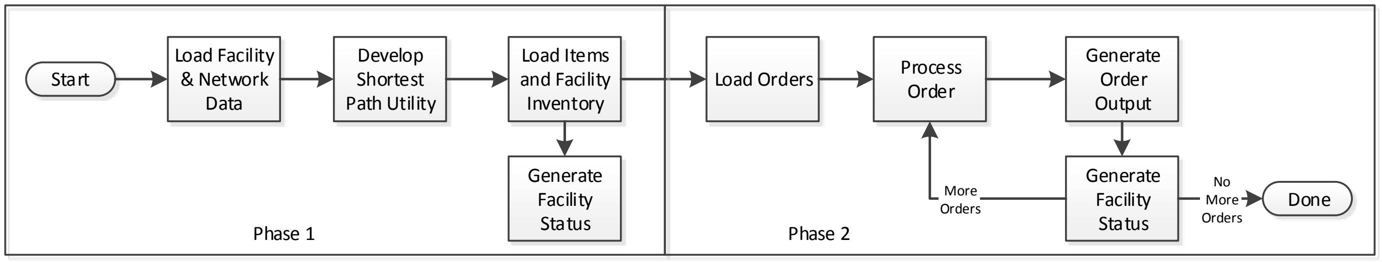
- Facility Inventory

Facility	Item Id (Quantity)				
Seattle, WA	ABC123 (45)	JBL3100 (52)	PL132-C (4)	PU238 (60)	
San Francisco, CA	RL123A (40)	XLK200B (8)	SR71-D (81)	CT1928 (46)	
Los Angeles, CA	RX100-3 (57)	JBL3100 (60)	E241i (44)	XTP202 (32)	
Phoenix, AZ	RX100-3 (18)	SR71-D (55)	SN-241-L (22)	ZTF109 (54)	
Denver, CO	ABC123 (80)	RL123A (110)	RTF110 (100)		
Santa Fe, NM	RX100-3 (144)	CR2032 (280)			
Fargo, ND	ABC123 (25)	JBL3100 (32)	XLK200B (82)		
Austin, TX	RL123A (66)	RX100-3 (35)	SN-241-L (35)		
St. Louis, MO	ABC123 (70)	SR71-D (45)	PU238 (82)		
Chicago, IL	ABC123 (60)	E241i (64)	CT1928 (20)	RTF110 (110)	XTP202 (20)
New Orleans, LA	MM35P (100)	PU238 (49)	ZTF109 (120)		
Nashville, TN	RX100-3 (78)	XLK200B (28)	CR2032 (366)	SN-241-L (16)	
Detroit, MI	JBL3100 (117)	SR71-D (8)	CT1928 (72)		
Boston, MA	MM35P (47)	PL132-C (3)	CT1928 (65)		
New York City, NY	RL123A (12)	CR2032 (132)	E241i (72)	ZTF109 (28)	
Norfolk, VA	MM35P (66)	RL123A (50)	XLK200B (14)		
Atlanta, GA	RX100-3 (90)	E241i (32)	PU238 (120)		
Miami, FL	MM35P (16)	JBL3100 (44)	SR71-D (15)	SN-241-L (28)	XTP202 (88)

- Mandatory Orders (6)

Order Id: TO-001 Order Time: Day 1 Destination: Miami, FL Order Items: ▪ Item: ABC123, Qty: 180 ▪ Item: CR2032, Qty: 320	**Order Id: TO-002** Order Time: Day 1 Destination: Boston, MA Order Items: ▪ Item: PL132-C, Qty: 5 ▪ Item: XLK200B, Qty: 120	**Order Id: TO-003** Order Time: Day 4 Destination: Seattle, WA Order Items: ▪ Item: XTP202, Qty: 100 ▪ Item: E241i, Qty: 105
Order Id: TO-004 Order Time: Day 10 Destination: Phoenix, AZ Order Items: ▪ Item: CR2032, Qty: 200 ▪ Item: SR71-D, Qty: 85	**Order Id: TO-005** Order Time: Day 7 Destination: Chicago, IL Order Items: ▪ Item: XLK200B, Qty: 120 ▪ Item: RX100-3, Qty: 200	**Order Id: TO-006** Order Time: Day 8 Destination: Denver, CO Order Items: ▪ Item: CT1928, Qty: 120

Development Plan



- Implementation Part 1 (Start: 4/14 – Due: 5/5, 5:45pm Central Time)
  - Load Facilities, Inventories, Transport Network and Item catalog; Shortest-Path Determination; Generate Facility Output
  - In this phase, you must fully implement the following:
    - Create the Facility Inventory XML File (offline process)
    - Load the Facility Inventory XML File and create internal representation of Facility and its inventory
    - Create the Transport Network File (offline process)
    - Load the Transport Network File and create internal representation of Transport Network
    - Develop/research and implement algorithm to determine the shortest path between 2 Facilities (in decimal days)
    - Create the Items File (offline process)
    - Load the Items File and create internal representation of the Items
    - Generate Facility Status Output
  - Submission: ZIP'd project folder *with compiled .class & JAR files removed before ZIP'ing*. Late submissions will be penalized by 10% per week late. *Submissions made WITHOUT removing compiled .class & JAR files will be penalized.*

- Implementation Part 2 (Start: 5/5 – Due: 6/2, 5:45pm Central Time)
  - Load Orders, Process Orders, Generate Order Outputs, Generate Facility Outputs
  - In this phase, you must fully implement the following:
    - Create the Order File (offline process)
    - Load the Order File and create internal representation of Orders & Related Order Items
    - Fully implement Order Processing functionality
    - Generate Order Output
  - Submission: ZIP'd project folder with compiled .class & JAR files removed before ZIP'ing. NO Late submissions allowed. *Submissions made WITHOUT removing compiled .class & JAR files will be penalized.*

Exceptions & Exception Handling

It is important to note that any public method that accepts parameters should validate the incoming parameters before using them.

If bad/unusable or invalid (by application standards) data is detected, an appropriate exception should be thrown.

You should create your own exception classes to handle all such cases, *do not use "Exception" or any pre-existing java exception classes*. Remember to only "catch" an exception where the problem encountered can be "fixed". Likewise, methods that access system resources should manage any related exceptions in an orderly manner.

Remember that if you decide to propagate any exceptions from methods that are declared in an interface, the interface declaration of that method must also indicate that the same exceptions are thrown.

Project Assistance

If you are stuck on some design or code related problem that you have exhaustively researched and/or debugged yourself, you can email me a ZIP file of your entire project so that I can examine the problem. All emailed assistance requests must include a detailed description of the problem, and the details of what steps you have already taken in trying to determine the source of the problem.

Submissions & Grading

All submissions must include all required design documents or code necessary to compile and run your application. Submitted code must be in the original package-folder form. ZIP'd project folder submissions are suitable **(*with compiled .class & JAR files removed before ZIP'ing*)**.

The following are the key points that will be examined in Project when graded:

- Good Object Oriented Design & Implementation
- Good Programming Style & Appearance
- Properly formatted, useful Javadoc documentation
- Properly written, JUnit tests with good code coverage
- Proper Application Execution & Output

When submitting, you should submit a ZIP file of your entire project (***with compiled .class & JAR files removed before ZIP'ing***) so that I can compile and execute it on my end.

Late submissions will be penalized by 10% per week late. (i.e., from one second late to 1 week late: 10% penalty, from one week plus one second late to 2 weeks late: 20% penalty, etc.). No late submissions are allowed for the final (2nd) phase.

NO submissions of any assignment will be accepted after class time on Week 10 (6/2).

If you do not understand anything in this handout, please ask. Otherwise the assumption is that you understand the content.