

# KawrgoJumper

## PROJECT PITCH

Pitch Lead: Kareem Shehada

Team Members: Ruben Ruiz, Padma Bellamkonda,  
Jaydah Julian, Win Thant

Date: 10/28/2024



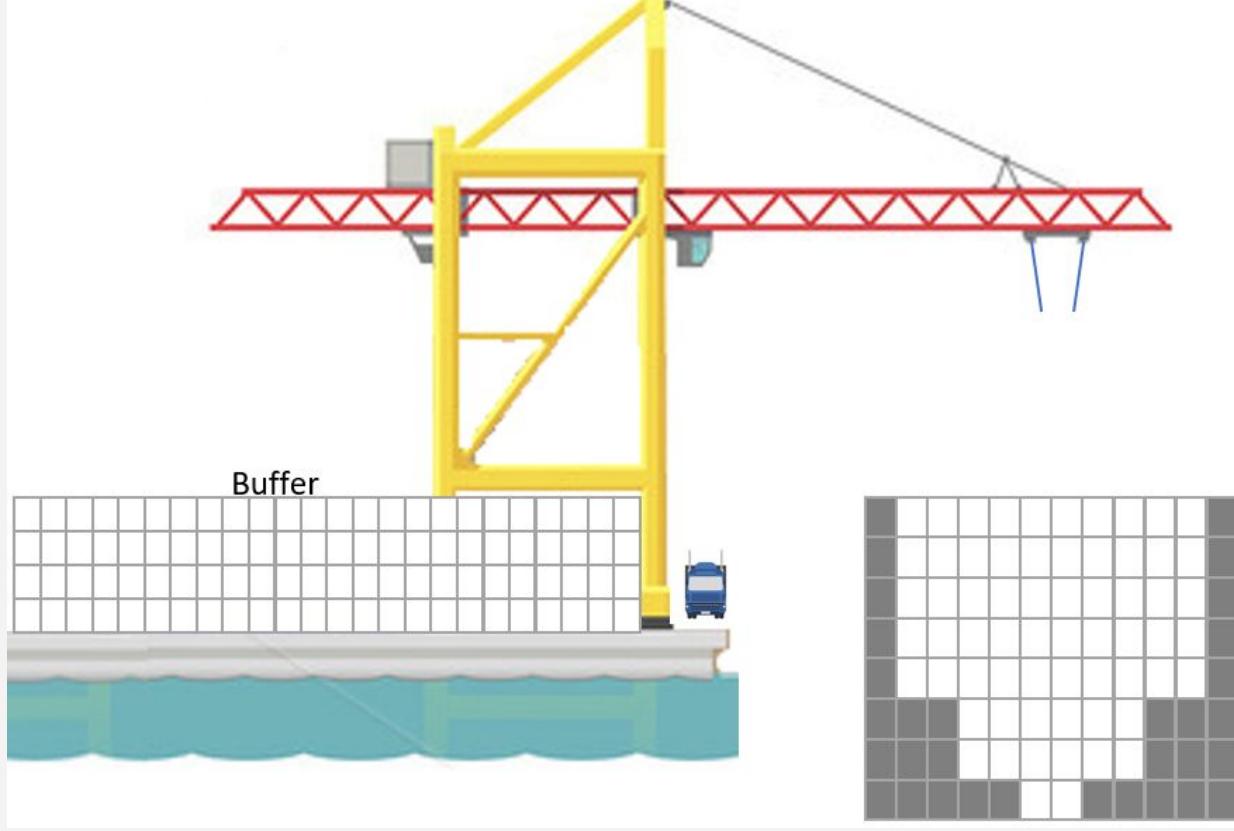
**KawrgoJumper**  
**SHIPPING LOGISTICS**

# OUR BASIC UNDERSTANDING



You want a program that can handle two tasks in optimal time, one at a time:

- Unloading/loading containers from a ship.
- Legally balance a ship by moving containers.



# Figure 1

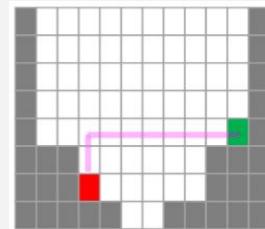
# General Understanding

General understanding applicable to both problems:

- The maximum ship size will be 8 rows by 12 columns. [a]
- Containers may be placed on top of row 8 (in row 9 & 10), but the ship cannot leave with them there. [b]
- There is a buffer of size 96, 4 rows by 24 columns. [c]
- The layout of the port as shown in Figure 1: Buffer, Truck Loading Area, Ship [d]
  - Assume this layout & perspective for all future points and assumptions.

# General Understanding

- The cost to move one cell within the buffer or ship is 1 minute. [e]
  - This is also the Manhattan distance.
  - The manhattan distance is the distance between two points on a grid, without taking diagonals.
  - Ex: Moving from red to green slot will take 9 minutes.



- The cost to move from the top left edge of the ship to the top right edge of the buffer is 4 minutes, & vice versa.
  - Plus time to move the container within the ship or the buffer.[f]

# General Understanding

General understanding applicable to both problems:

- Our program will
  - Read a structured manifest.
  - Generate an order to move containers to complete the task in **optimal** time.
  - Write a log of all instructions.
  - Accept operator comments to the log.
  - Not allow direct operator editing of the log.
  - Begin writing to a new log when a new year starts.
  - Support a maximum of 96 containers on a ship, in smaller ships, this number will be smaller as indicated in the manifest.
  - Show a loading wheel during calculations so the operator does not mistake it for a crash.

# LOADING UNLOADING

- The program must accept operator input for the transfer list, which are containers that need to be loaded and unloaded.
  - *The transfer list is never directly read by our program* [a]
- The number of containers that can be unloaded will be from 0 to the number of containers on the ship. [b]
- The number of containers that can be loaded will be from 0 to 96. [c]
- The program must error if the number of containers to be loaded will exceed the remaining space after unloading. [d]
- The cost to move to the truck from the closest edge of the buffer or ship is 2 minutes, & vice versa. [e]

# LOADING UNLOADING

Our program will follow these steps at a high level:

- 1) Ask operator to login (if not already).
- 2) Ask operator to upload the manifest ← To know what is on the ship
- 3) Ask operator what is going to be unloaded.
- 4) Ask operator how many containers will be loaded.
  - a) If the amount to load is greater than the UNUSED slots on the ship plus the amount to unload, then error.
- 5) Provide steps for unloading containers.
- 6) Provide steps for loading containers.
  - a) Additionally ask for container name when it's being loaded.
  - b) Ask for container weight
- 7) Save updated manifest to Desktop

# Balancing

- We may use custom algorithms to determine the legally balanced order that is fastest to achieve.
- If these fail or it is not possible, we may resort to the SIFT result and generate optimal steps to reach it.
  - The SIFT result is legally balanced, regardless of actual weight on either side.
- A ship with one container is legally balanced

# Balancing

Once the operator chooses the Balancing option,  
Our program will follow these steps at a high level:

- 1) Ask current operator to login (if not already).
- 2) Ask operator to upload the manifest ← To know what is on the ship
- 3) Provide steps for balancing the ship.
- 4) Save updated manifest to Desktop.

# STAKEHOLDERS

- Crane Operators
- Port Management (Mr Keogh)
- Truck Drivers
- Head Office/Logistics Team
- Ship Captains/Crew
- Customs Officials
- Law Enforcement
- Insurance Company
- Unions
- Safety Inspectors / Compliance Officers
- Shipping Companies / Container Owners
- Local Residents / Business Owners

# DETAILED WORKFLOW

Here is our understanding of an example workflow. [a]

- The operator in the cabin receives an email from the incoming ship.
  - This happens at least 15 minutes before the ship is docked.
  - This email contains the manifest.
- The operator **will** save this manifest and place it in the desktop.
- The operator will also receive a transfer list from the head office, which contains specifics about which containers need to be unloaded and loaded onto the ship.
  - This is in no particular order.
  - This **will** come in many different ways: email, fax, phone call, text, etc.



# DETAILED WORKFLOW



- The Operator checks to see if the required trucks are waiting in the holding area. If a truck is missing, they can make a few calls to check but generally,
  - For **Unloading**: the operator calls the Port of Long Beach for special truck to take container to a special location.
  - For **Loading**: the operator can consider that this load will not be taking place, and that the ship is sailing without this container.
- By this point, the operator knows exactly what is going on and off the ship.

# DETAILED WORKFLOW



- Our program is on the operator's browser, and after logging in, it displays an upload menu for the manifest.
- Once the manifest is uploaded, the program knows exactly what is on the ship.
- Additionally, once uploaded two options are shown as buttons:  
**Unload/Load** and **Balance**



# DETAILED WORKFLOW

- If the operator clicks on the **Unload/Load** button, the system will ask the operator to input what **container** is **being unloaded** from the ship.
- The system then asks the operator to input the **number** of containers **being loaded** on the ship.
  - Once again, this can be from 0 to 96 container
  - If the number of containers loaded is 0 (no containers loaded) operation move to moving the specified unloaded containers
- At this point, the system can calculate the optimal container order, as it knows exactly what containers will be **loaded/unloaded**.
- The weight of the containers being loaded will not be known yet as they have not been picked up.

# DETAILED WORKFLOW



- The program will have **exactly** 15 minutes of free time to compute the optimal order.
- Once a solution is calculated, the program tells the operator a solution is ready as well as the estimated time it will take to perform all the operations.
- The program provides a step by step instruction to the operator.
  - The operator will click “Next Step” or similar button after performing each step.
  - This is to avoid missteps by human error.
  - The last step always has to be the “park position” for the crane

# DETAILED WORKFLOW



- Once the final step is done, the program assumes the process is complete.
- It will push to the Desktop an updated version of the manifest named:

<shipname>OUTBOUND.txt

- The program will inform the operator that this has been done, and also remind them to email this manifest file to the ship.
- Once emailed, the operator can dismiss this reminder, through a button.
- The program will then return back to its initial screen to **Upload Manifest.**

# ASSUMPTIONS

- Manifest is always correct <sup>[a]</sup>
- No need to consider damages to containers for our program, assume possible and legal moves are always achievable. <sup>[b]</sup>
- Operators will follow all the steps down to the letter. <sup>[c]</sup>
- Manifest saved to the desktop from the captain's email is perfect and readable 100 percent. <sup>[d]</sup>
- Operators will click for next steps the moment a task/step is completed. <sup>[e]</sup>
- The PCs that will be used will have same/similar specifications to the cheapest COSTCO all-in-one PC. <sup>[f]</sup>
- Mr Keogh has exemption from American Disabilities Act: All employees are able-bodied, not deaf, have perfect vision, & are not color blind <sup>[g]</sup>

# INPUTS

- Time from an NTP server (Network Time Protocol)
  - time.nist.gov
  - Government source
  - Payload interpreted by library.
  - Formatted according to Time Regulations.
- Containers to unload from operator
  - Selected in grid menu
- Number of containers to load from operator
  - Integer from 0 to 96
- New container name from operator
  - String, likely UTF encoded
  - First character needs to be a principal character [a]
  - Max length of container name is 255 characters [b]
  - Must be converted to ASCII
- New container weight from operator
  - Five digit integer from 0 to 99,999

# INPUTS

- Operator username
  - String on login screen
- Operator comments
  - Arbitrary text
  - Write to log file
- Operator button clicks
  - Mouse button 1 down & up on program buttons
- Manifest (Details in next slide)

# Inputs: Manifest

Manifest uploaded by Operator

- The manifest is 100% correct.
- The manifest contains 96 entries of ASCII encoded text, in the following format:



- If an entry corresponds to an inaccessible cell on the ship, its name will be **NAN**. [g]
- If an entry corresponds to an unused cell on the ship, its name will be **UNUSED**. [h]
- It is possible for there to be duplicate container names.
- The weight cannot exceed 99,999 kilos, in practice it will be much less. [i]

# OUTPUTS

- Updated Manifest
  - Exact same format as inputted manifest.
  - Name is <manifestName>OUTBOUND.txt
  - Saved to Desktop
- Currently selected manifest name
  - Text on screen: "Current File: QueenVictoria.txt"
- Current job
  - Text on screen
  - Either "Load/Unload" or "Balance"
- Container movement steps
  - One step at a time
  - "Current Task: Load container (Apples) to 5,2 on ship."
  - **Display estimated time (ETA) on screen**
- Manifest interpreted as a grid
  - NAN as dark color, UNUSED as white, used as blue, selected green
  - May include coordinate [row,col], container name, and weight in grid cell.

# OUTPUTS

- Estimated time to finish
  - “ETA: 10min”
  - Calculated by counting area that must be crossed for all steps.
- Error messages
  - Red text, clearly explain what's wrong
  - “Error: Cannot load 4, space left is 2”
- Done message
  - Contains output manifest file name
  - Contains reminder to email captain
- Logs (more info next slide)

# Outputs: Logs

- When the program is opened it will search for KeoghPort<Year> in a fixed directory.
  - If found, write logs to KeoghsPort<Year>.txt
  - If not found, create the file KeoghsPort<Year>.txt
- When a new year is detected, the program will create a new log file KeoghsPort<Year>.txt [j]
- Every atomic event gets a new line beginning with the date and time, to the nearest minute.
- Event format is, for example: "2024-01-05 15:12 John Smith has signed in"
- The log file will use ASCII encoding
- More info on time regulation later.

# SCENARIO I: PART 1 of 10

- John Smith is a 5-year employee of Mr. Keogh
- John is a crane operator that works the night shift (12 to 8)
- John arrives at the port at around 11:45 pm (23:45) on July 26th 2025
- He begins his shift by logging in at exactly 12:01 am (00:01)

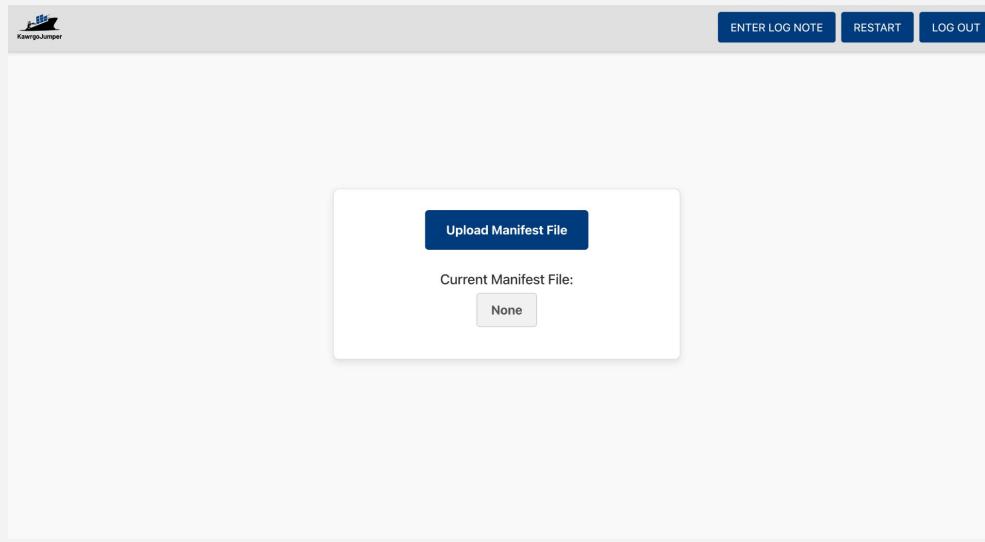


USERNAME

Log In

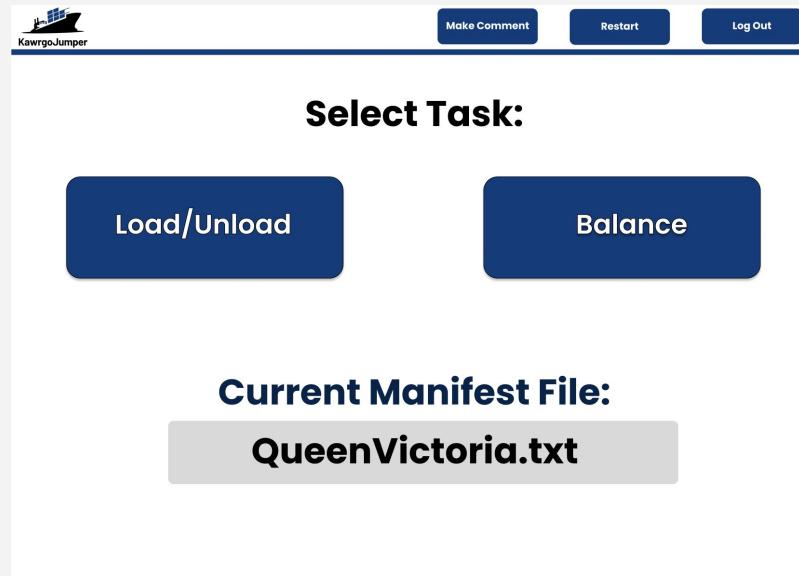
# SCENARIO I: PART 2 of 10

- At around 1:15 am (01:15), the head office informs him that a ship is expected to arrive at the port in about an hour.
- At 2:05 am (02:05), the Queen Victoria emails their manifest to John.
- He downloads the manifest on to the desktop and uploads it to the program



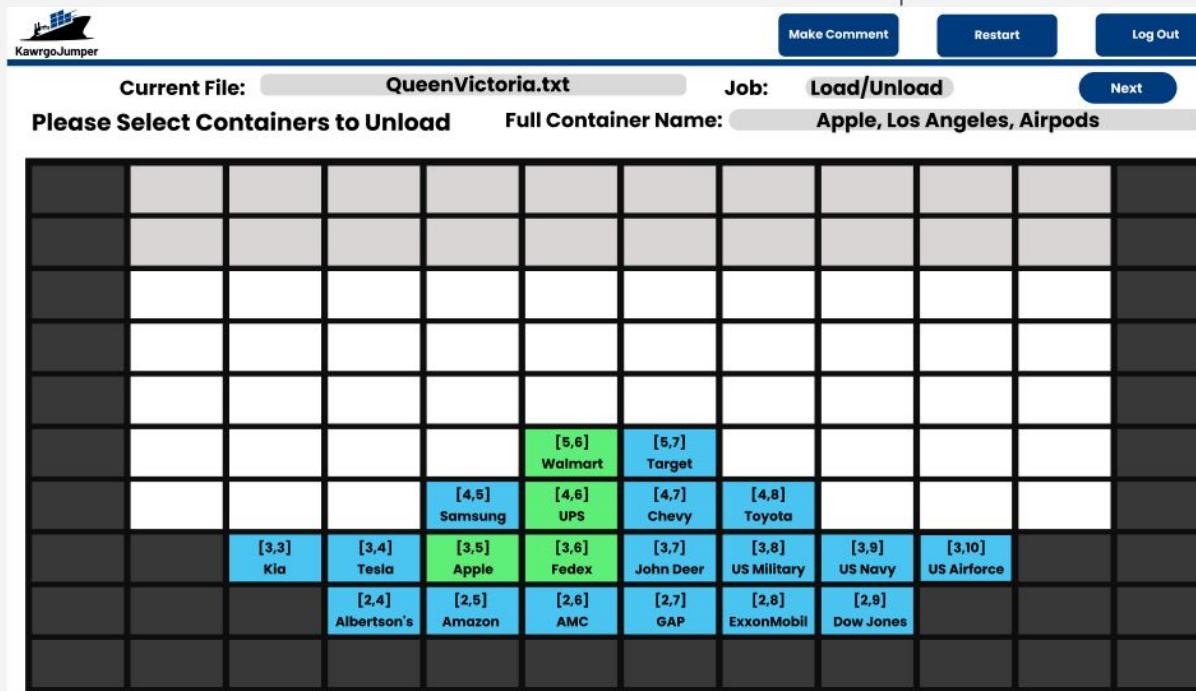
# SCENARIO I: PART 3 of 10

- At 2:07 am (02:07), the head office emails him the transfer list.
- At 2:20 am (02:20), the ship docks.
- The transfer list details all the containers that need to be unloaded and loaded.
- John selects the Load/Unload task



# **SCENARIO I: PART 4 of 10**

- John then selects all the containers listed for unload in the transfer list and clicks begin.



# SCENARIO I: PART 5 of 10

- John then enters the containers to be loaded

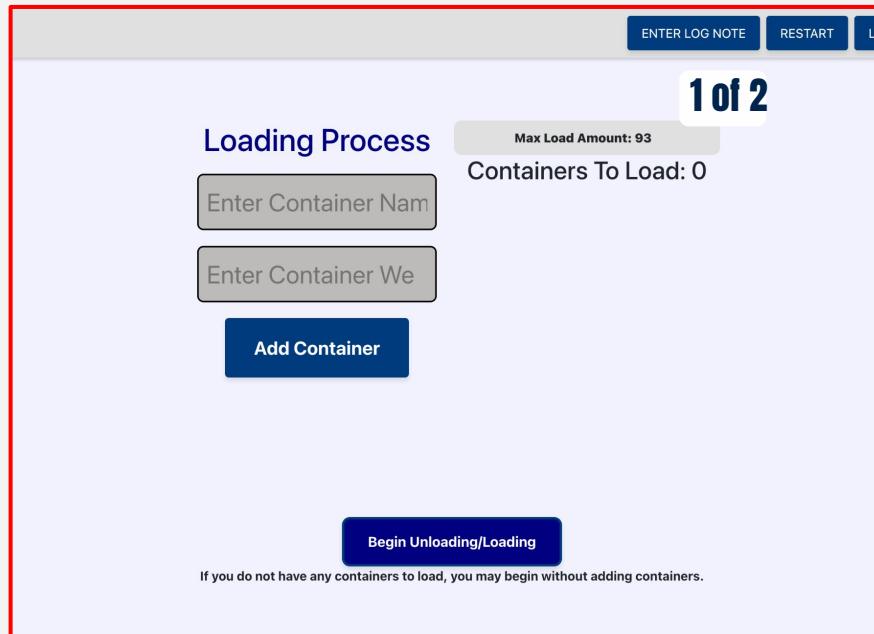
The screenshot shows a user interface for managing shipping containers. At the top right are three buttons: 'ENTER LOG NOTE' (white text on dark blue), 'RESTART' (white text on dark blue), and 'LOG OUT' (white text on dark blue). Below these is a section titled 'Loading Process' with a logo of a cargo ship. To the right, it displays 'Max Load Amount: 93' and 'Containers To Load: 3'. Three items are listed with 'Delete' buttons: 'cont 1 - 2kg', 'cont 2 - 4kg', and 'cont 3 - 6kg'. On the left, there are two large rectangular boxes: one containing 'cont 3' and another containing the number '6'. At the bottom left is a blue button labeled 'Add Container'. At the bottom center is a blue button labeled 'Begin Unloading>Loading'. A message at the bottom states: 'If you do not have any containers to load, you may begin without adding containers.'

# **SCENARIO I: PART 6 of 10**

- The program tells him which container to move, it repeats this until all containers are unloaded.

# SCENARIO I: PART 7 of 10

- John then signals for one of the trucks on standby to approach the transfer location.
- The program then instructs John to pick up the container from the truck and to enter its name and weight.



# SCENARIO I: PART 8 of 10

- The program instructs John on where to place the new container.

The screenshot shows a software interface for managing a ship's manifest. At the top, there are navigation buttons: a plus sign icon on the left, the title "SCENARIO I: PART 8 of 10" in large bold letters, and a plus sign icon on the right. Below the title, there are three buttons: "Make Comment", "Restart", and "Log Out".

The main area displays the following information:

- Current File:** QueenVictoria.txt
- Manifest Ship Name:** Queen Victoria
- Job:** Load/Unload
- ETA:** 20min
- Current Task:** Load container (Circus Peanuts) to 4,2 (in Green)

Below this, there are two buttons: "Next Step" and "Show Buffer".

The central feature is a 10x10 grid representing the ship's deck. The columns are labeled 1 through 10 and the rows are labeled 1 through 10. The grid contains several colored cells representing different types of containers:

- Target:** A blue cell at position [5,7] labeled "Target".
- Circus Peanuts:** A green cell at position [4,2] labeled "[4,2] Circus Pea..".
- Samsung:** A light blue cell at position [4,5] labeled "[4,5] Samsung".
- Kia:** A light blue cell at position [3,3] labeled "[3,3] Kia".
- Tesla:** A light blue cell at position [3,4] labeled "[3,4] Tesla".
- Albertson's:** A light blue cell at position [2,4] labeled "[2,4] Albertson's".
- Amazon:** A light blue cell at position [2,5] labeled "[2,5] Amazon".
- AMC:** A light blue cell at position [2,6] labeled "[2,6] AMC".
- GAP:** A light blue cell at position [2,7] labeled "[2,7] GAP".
- ExxonMobil:** A light blue cell at position [2,8] labeled "[2,8] ExxonMobil".
- Dow Jones:** A light blue cell at position [2,9] labeled "[2,9] Dow Jones".
- US Military:** A light blue cell at position [3,8] labeled "[3,8] US Military".
- US Navy:** A light blue cell at position [3,9] labeled "[3,9] US Navy".
- US Airforce:** A light blue cell at position [3,10] labeled "[3,10] US Airforce".

# SCENARIO I: PART 9 of 10

- This process repeats until all new containers are loaded.



**Containers left to load: 1**

**Please pick up the next container to load  
and fill out the following:**

Enter Container Name to load

Tardigrades

Enter Container Weight

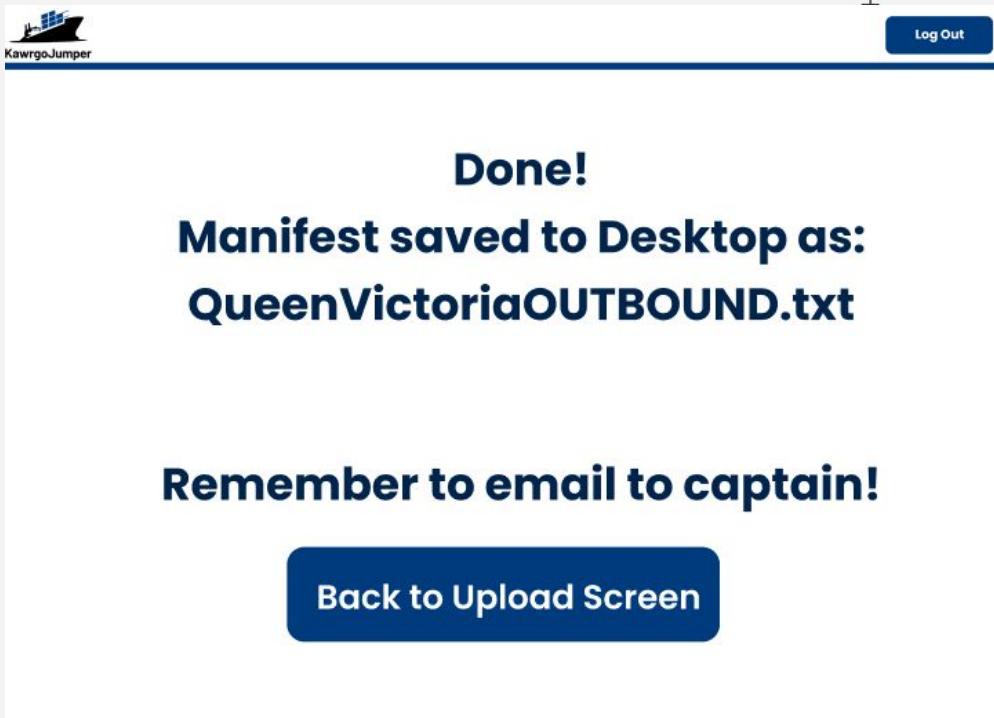
10

**Next**

Log Out

# SCENARIO I: PART 10 of 10

- Once all containers are loaded and unloaded, the manifest is saved to the Desktop.
- John emails the updated manifest to the captain of the Queen Victoria.



The screenshot shows a software application window titled "Kawrgo-Jumper". At the top left is a small icon of a ship, and at the top right is a "Log Out" button. The main content area displays the message "Done!" in large blue text, followed by "Manifest saved to Desktop as:" and the file name "QueenVictoriaOUTBOUND.txt" in bold blue text. Below this, a large blue button with white text reads "Back to Upload Screen".

**Done!**

**Manifest saved to Desktop as:**

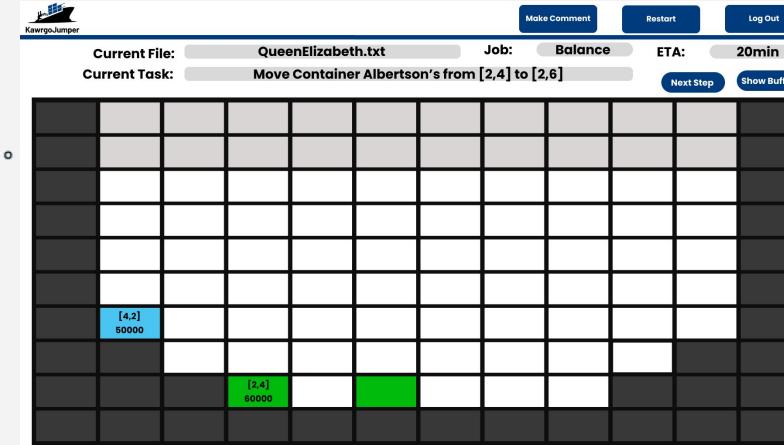
**QueenVictoriaOUTBOUND.txt**

**Remember to email to captain!**

**Back to Upload Screen**

# SCENARIO II: PART 1 of 6

- Amy Christopher is a 6-month employee of Mr. Keogh
- Amy is a crane operator that works the shift after John, from 8 am to 4 pm.
- Amy arrives at the port at around 7:45 am on January 16, 2025
- She replaces John, who was working the previous shift while a task to balance a container is ongoing on the program.
- John logs out, gets up, and leaves, at exactly 8:01 am. The current task is saved by the program and the logout is logged.



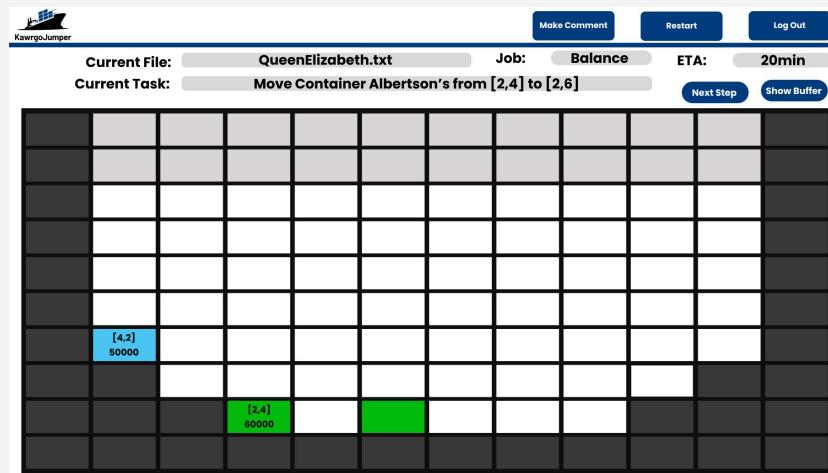
# SCENARIO II: PART 2 of 6

- Amy is now at the Login page, where she logs in with her username.
- Once logged in, the program records her login at 8:01 am.



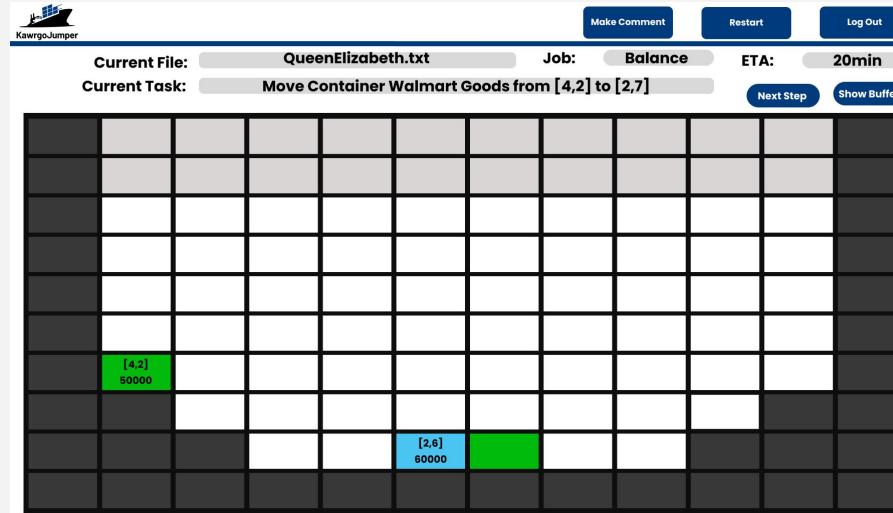
# SCENARIO II: PART 3 of 6

- The previous task which was ongoing during John's shift is once again displayed to Amy, who continues at the step where John left off.
- After moving the container to the green slot highlighted, Amy clicks on the Next Step.
- This is recorded in the log.



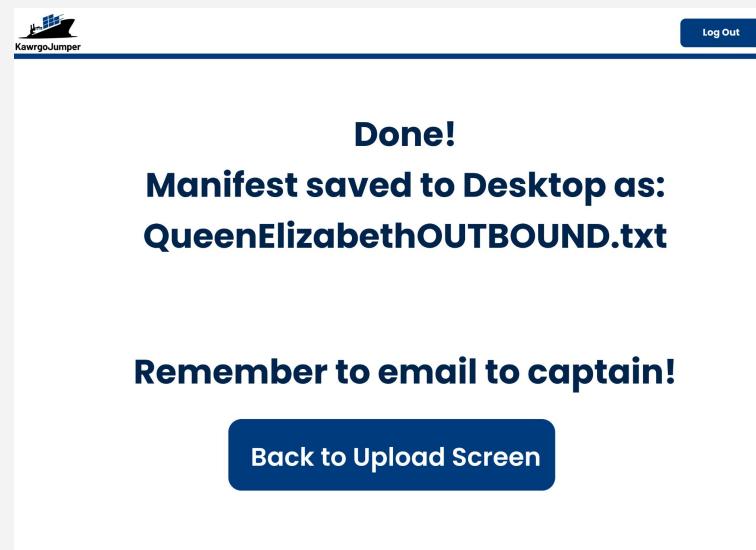
# SCENARIO II: PART 4 of 6

- The Next Step tells Amy to move the other container, which is now highlighted in Green.
- Once moved, she clicks on Next Step again.
- This is recorded in the log.



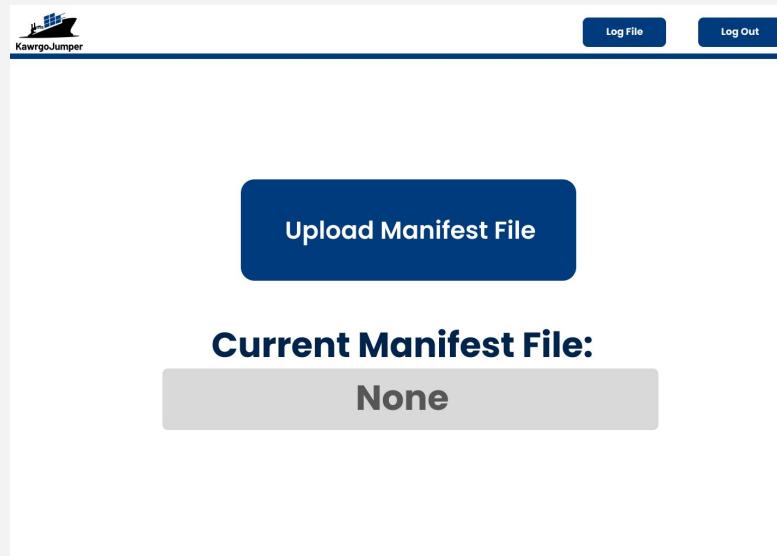
# SCENARIO II: PART 5 of 6

- The balance operation has finished, and the program saves the updated manifest to the Desktop.
- The program displays a screen to tell Amy that it is complete.
- It also reminds Amy to email the captain the new manifest.



# SCENARIO II: PART 6 of 6

- After emailing the new manifest, Amy clicks on the “Back to Upload Screen”, which takes her back to the part where she can upload a manifest file for her next job.



# SCENARIO III: PART 1 of 9

- Jalen Rose is a 10-year employee of Mr. Keogh
- Jalen is a crane operator that works the shift from 4 pm to 12 midnight.
- Jalen arrives at the port at around 3:45 pm (15:45) on February 26th 2025
- He takes over for another employee and begins his shift by logging in at exactly 4:01 am (16:01)

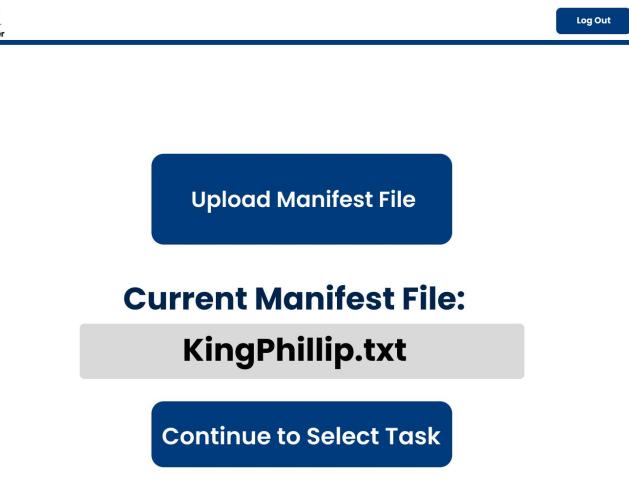
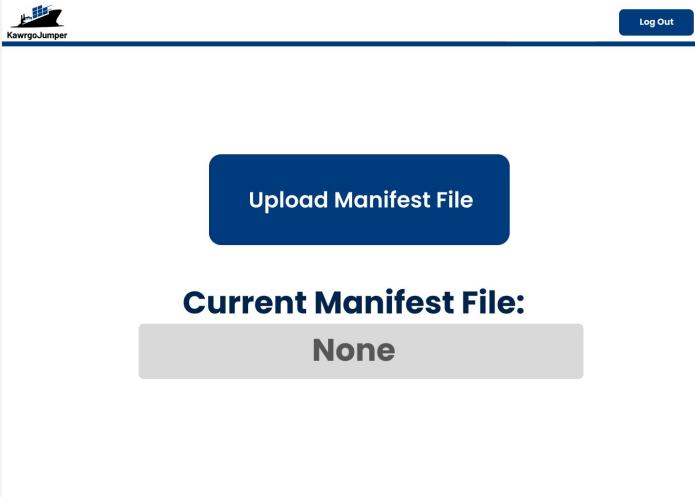


USERNAME

Log In

# SCENARIO III: PART 2 of 9

- At around 6:15 pm (18:15), the head office informs him that a ship is expected to arrive at the port in about an hour.
- At 7:05 pm (19:05), the captain of King Edward emails their manifest to Jalen.
- He downloads the manifest onto the desktop and uploads it to the program.



# SCENARIO III: PART 3 of 9

- At 7:07 pm (02:07), the head office emails him the transfer list.
- At 7:20 pm (02:20), the ship docks.
- The transfer list details all the containers that need to be unloaded and loaded.
- Before Jalen clicks on the Load/Unload option, he notices he has uploaded the wrong manifest. He clicks on the Restart button, which takes him back to the screen to upload the correct manifest.

The image displays two side-by-side screenshots of a software application interface. Both screenshots feature a header with a ship icon and the text 'KawgoJumper' on the left, and three buttons: 'Make Comment', 'Restart', and 'Log Out' on the right.

**Screenshot 1 (Left):** The title 'Select Task:' is displayed above two large blue buttons labeled 'Load/Unload' and 'Balance'. Below this, the section 'Current Manifest File:' shows a grey button with the text 'KingPhillip.txt'.

**Screenshot 2 (Right):** The title 'Select Task:' is partially visible at the top. A single large blue button labeled 'Upload Manifest File' is centered. Below it, the section 'Current Manifest File:' shows a grey button with the text 'None'.

# SCENARIO III: PART 4 of 9

- Jalen then selects all the containers listed for unload in the transfer list and clicks next.
- For this task, Jalen only has to unload 1 container, called Kia San Francisco Tire Parts, which he selects.

Please Select Containers to Unload											
Full Container Name: Kia San Francisco Tire Parts											
	[8,2] Apple, Los..	[8,3] Kia	[8,4] Tesla	[8,5] Apple, Los...	[8,6] Fedex	[8,7] John Deer	[8,8] US Military	[8,9] US Navy	[8,10] US Airforce	[8,11] Kia	
	[7,2] Kia	[7,3] Kia	[7,4] Tesla	[7,5] Apple, Los...	[7,6] Fedex	[7,7] John Deer	[7,8] US Military	[7,9] US Navy	[7,10] US Airforce	[7,11] Kia	
	[6,2] Kia	[6,3] Kia	[6,4] Tesla	[6,5] Apple, Los...	[6,6] Fedex	[6,7] John Deer	[6,8] US Military	[6,9] US Navy	[6,10] US Airforce	[6,11] Kia	
	[5,2] Kia	[5,3] Kia	[5,4] Kia	[5,5] Kia	[5,6] Walmart	[5,7] Target	[5,8] US Military	[5,9] US Navy	[5,10] US Airforce	[5,11] Kia	
	[4,2] US Military	[4,3] US Navy	[4,4] US Airforce	[4,5] Samsung	[4,6] UPS	[4,7] Chevy	[4,8] Toyota	[4,9] Kia	[4,10] Kia	[4,11] Kia	
		[3,3] Kia	[3,4] Tesla	[3,5] Apple, Los...	[3,6] Fedex	[3,7] John Deer	[3,8] US Military	[3,9] US Navy	[3,10] US Airforce		
			[2,4] Albertson's	[2,5] Amazon	[2,6] AMC	[2,7] GAP	[2,8] ExxonMobil	[2,9] Dow Jones			

# SCENARIO III: PART 5 of 9

- Jalen then enters the containers to be loaded
- Jalen also only has to load 1 container.
- Jalen mistakenly enters 2 as the number. The system checks the manifest and sees that only one slot is free, and Jalen can delete the extra container added.

The screenshot shows a software interface for managing shipping containers. At the top, there are three buttons: 'ENTER LOG NOTE' (blue), 'RESTART' (white), and 'LOG OUT' (blue). Below this, the title 'Loading Process' is displayed in blue. A message 'Max Load Amount: 93' is shown above a field labeled 'Containers To Load: 0'. There are two input fields: 'Enter Container Nam' and 'Enter Container We'. A large blue button labeled 'Add Container' is at the bottom. In the center, there is a yellow gradient background.

**Left Panel (Initial State):**

- Max Load Amount: 93
- Containers To Load: 0
- Enter Container Nam
- Enter Container We
- Add Container

**Right Panel (After Mistake):**

- Max Load Amount: 93
- Containers To Load: 2
- cont2
- 2
- cont1 - 1kg Delete
- cont2 - 2kg Delete
- Add Container

**Buttons at the Bottom:**

- Begin Unloading>Loading
- If you do not have any containers to load, you may begin without adding containers.

# SCENARIO III: PART 6 of 9

- Jalen realizes the error, and inputs the correct number this time.
- The program verifies the load amount is possible & it displays the steps to first unload the Kia container that Jalen previously selected.

KawgoJumper

Current File:		KingEdward.txt		Job:	Load/Unload	ETA:	5min	Make Comment		Restart		Log Out	
Current Task:		Move container in 8,3 (Kia) to truck						<a href="#">Next Step</a>	<a href="#">Show Buffer</a>				
[8,2] Apple, Los..	[8,3] Kia	[8,4] Tesla	[8,5] Apple, Los..	[8,6] Fedex	[8,7] John Deer	[8,8] US Military	[8,9] US Navy	[8,10] US Airforce	[8,11] Kia				
[7,2] Kia	[7,3] Kia	[7,4] Tesla	[7,5] Apple, Los..	[7,6] Fedex	[7,7] John Deer	[7,8] US Military	[7,9] US Navy	[7,10] US Airforce	[7,11] Kia				
[6,2] Kia	[6,3] Kia	[6,4] Tesla	[6,5] Apple, Los..	[6,6] Fedex	[6,7] John Deer	[6,8] US Military	[6,9] US Navy	[6,10] US Airforce	[6,11] Kia				
[5,2] Kia	[5,3] Kia	[5,4] Tesla	[5,5] Apple, Los..	[5,6] Fedex	[5,7] John Deer	[5,8] US Military	[5,9] US Navy	[5,10] US Airforce	[5,11] Kia				
[4,2] US Military	[4,3] US Navy	[4,4] US Airforce	[4,5] Samsung	[4,6] UPS	[4,7] Chevy	[4,8] Toyota	[4,9] Kia	[4,10] Kia	[4,11] Kia				
[3,2] Kia	[3,3] Tesla	[3,4] Apple, Los..	[3,5] Fedex	[3,6] John Deer	[3,7] US Military	[3,8] US Navy	[3,9] US Airforce						
		[2,4] Albertson's	[2,5] Amazon	[2,6] AMC	[2,7] GAP	[2,8] ExxonMobil	[2,9] Dow Jones						

KawgoJumper

Current File:		KingEdward.txt		Job:	Load/Unload	ETA:	5min	Make Comment		Restart		Log Out	
Current Task:		Move container in 8,3 (Kia) to truck						<a href="#">Next Step</a>	<a href="#">Show Buffer</a>				
[8,2] Apple, Los..		[8,4] Tesla	[8,5] Apple, Los..	[8,6] Fedex	[8,7] John Deer	[8,8] US Military	[8,9] US Navy	[8,10] US Airforce	[8,11] Kia				
[7,2] Kia	[7,3] Kia	[7,4] Tesla	[7,5] Apple, Los..	[7,6] Fedex	[7,7] John Deer	[7,8] US Military	[7,9] US Navy	[7,10] US Airforce	[7,11] Kia				
[6,2] Kia	[6,3] Kia	[6,4] Tesla	[6,5] Apple, Los..	[6,6] Fedex	[6,7] John Deer	[6,8] US Military	[6,9] US Navy	[6,10] US Airforce	[6,11] Kia				
[5,2] Kia	[5,3] Kia	[5,4] Tesla	[5,5] Apple, Los..	[5,6] Fedex	[5,7] John Deer	[5,8] US Military	[5,9] US Navy	[5,10] US Airforce	[5,11] Kia				
[4,2] US Military	[4,3] US Navy	[4,4] US Airforce	[4,5] Samsung	[4,6] UPS	[4,7] Chevy	[4,8] Toyota	[4,9] Kia	[4,10] Kia	[4,11] Kia				
[3,2] Kia	[3,3] Tesla	[3,4] Apple, Los..	[3,5] Fedex	[3,6] John Deer	[3,7] US Military	[3,8] US Navy	[3,9] US Airforce						
		[2,4] Albertson's	[2,5] Amazon	[2,6] AMC	[2,7] GAP	[2,8] ExxonMobil	[2,9] Dow Jones						

# SCENARIO III: PART 7 of 9

- After finishing the **unloading** of the container, Jalen then signals for one of the trucks on standby with the container to be loaded to approach the transfer location.
- The program then instructs John to pick up the container from the truck and to enter its name and weight.



Log Out

Containers left to load: 1

Please pick up the next container to load and fill out the following:

Enter Container Name to load  
Circus Peanuts

Enter Container Weight  
4555

Next

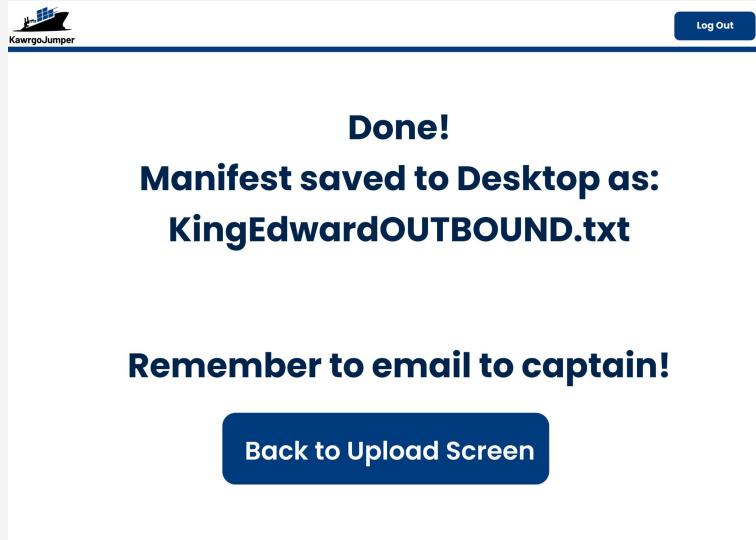
# SCENARIO III: PART 8 of 9

- The system displays the steps to load the Circus Peanuts container that Jalen previously selected.

QueenVictoria.txt												Make Comment	Restart	Log Out
Load container (Circus Peanuts) to 8,2 (in Green)												Next Step	Show Buffer	
[8,2] Apple, Los...	[8,3] Cicus Pean...	[8,4] Tesla	[8,5] Apple, Los...	[8,6] Fedex	[8,7] John Deer	[8,8] US Military	[8,9] US Navy	[8,10] US Airforce	[8,11] Kia					
[7,2] Kia	[7,3] Kia	[7,4] Tesla	[7,5] Apple, Los...	[7,6] Fedex	[7,7] John Deer	[7,8] US Military	[7,9] US Navy	[7,10] US Airforce	[7,11] Kia					
[6,2] Kia	[6,3] Kia	[6,4] Tesla	[6,5] Apple, Los...	[6,6] Fedex	[6,7] John Deer	[6,8] US Military	[6,9] US Navy	[6,10] US Airforce	[6,11] Kia					
[5,2] Kia	[5,3] Kia	[5,4] Kia	[5,5] Kia	[5,6] Walmart	[5,7] Target	[5,8] US Military	[5,9] US Navy	[5,10] US Airforce	[5,11] Kia					
[4,2] US Military	[4,3] US Navy	[4,4] US Airforce	[4,5] Samsung	[4,6] UPS	[4,7] Chevy	[4,8] Toyota	[4,9] Kia	[4,10] Kia	[4,11] Kia					
	[3,3] Kia	[3,4] Tesla	[3,5] Apple, Los...	[3,6] Fedex	[3,7] John Deer	[3,8] US Military	[3,9] US Navy	[3,10] US Airforce						
		[2,4] Albertson's	[2,5] Amazon	[2,6] AMC	[2,7] GAP	[2,8] ExxonMobil	[2,9] Dow Jones							

# SCENARIO III: PART 9 of 9

- After Jalen clicks next, the program displays the done screen, showing where the manifest is saved and reminding the user to email the captain.



# TIME REGULATIONS

- Time Formatting with Leading Zeros:
  - We'll use date formatting functions that guarantee leading zeros for any single-digit month, day, hour, or minute to ensure consistent readability and automated log parsing
  - The format applied will be **YYYY-MM-DD HH:MM** in 24-hour (military) time with a single space between DD and HH and a tab separating the timestamp from the log message
  - Instances where the log event occurs precisely on the hour
    - MM (minute) will display as 00.
- Pacific Standard Time (PST) with Daylight Savings (PDT):
  - All timestamps will be generated in PST or PDT
  - Program will automatically adjust for daylight saving time (PDT) using a time zone library like pytz (handles U.S. time zone changes automatically)
  - Will ensure consistent logging times across both standard and daylight saving time periods without requiring manual adjustments

2024-01-05 15:08

Manifest QueenMary opened, there are 19 containers on board

# TIME REGULATIONS

- Floor Function to Nearest Minute:
  - Used to truncate seconds
  - Rounding down to the nearest minute
  - Consistent and precise minute-by-minute logging ensured, even in cases where events happen near the end of one minute
- Maintaining Accurate Time:
  - Periodically check against an external network time protocol (NTP) server to ensure accurate time
    - Prevent major deviations, like seeing a log event timestamped in 1945 or 2099
    - Automatically log warnings in the system
    - Reset the program's internal clock to match the NTP server's time for log entries → overrides local machine
  - Local system clock that syncs upon network availability
  - Log time will default to the server time rather than the client (local) time

# UI CONSIDERATIONS



## KawrgoJumper

SHIPPING LOGISTICS

User Log In

USERNAME

Log In

# UI CONSIDERATIONS

The screenshot shows a user interface for a software application. At the top, there is a header bar with a logo of a ship and the text "KawgoJumper". On the right side of the header are three buttons: "ENTER LOG NOTE", "RESTART", and "LOG OUT". Below the header, a large red rectangular box highlights a central modal window. The modal has a white background and a thin gray border. Inside the modal, there is a large blue arrow pointing to the right, containing the word "Upload". To the right of this arrow is another blue button labeled "Upload Manifest File". Below these buttons, the text "Current Manifest File:" is displayed, followed by a button labeled "ShipCase5.txt". At the bottom of the modal, there is another blue button labeled "Continue to Select Task". A second blue arrow points to the left from the bottom right of the modal, containing the text "Select Manifest Name". The entire modal is centered on the page within the red box.

# UI CONSIDERATIONS



ENTER LOG NOTE

RESTART

LOG OUT

Select task:

Two Tasks

Load/Unload

Balance

Current Manifest:

ShipCase5.txt

# UI CONSIDERATIONS

The screenshot shows a user interface for managing shipping containers. At the top right, there's a logo for 'KawrgoJumper' featuring a stylized ship icon. To its right is a blue button labeled 'Buttons'. Below the logo are three grey buttons: 'ENTER LOG NOTE', 'RESTART', and 'LOG OUT'. On the left side, a blue callout box contains the text 'Easy-to-read fonts'. In the center, there are two grey status boxes: 'Current File: ShipCase5.txt' and 'Job: Load/Unload'. To the right of these is a blue button labeled 'Begin'. Below these elements is a heading 'Please Select Containers to Unload' followed by a large grid of 14x14 cells. Most cells are light grey and contain the word 'UNUSED'. A few cells are highlighted in green ('Dog', 'Pig') or blue ('Hen', 'Rat'). A red arrow points from the 'Contrasting Colors' callout at the bottom left to the colored cells. Another red arrow points from the 'Buttons' callout at the top right to the top navigation bar.

UNUSED													
UNUSED													
UNUSED													
UNUSED													
UNUSED													
UNUSED													
UNUSED	USED	UNUSED											

**Contrasting Colors**

# UI CONSIDERATIONS



KawrgoJumper

**ENTER LOG NOTE**

RESTART

[LOG OUT](#)

**Current File:** ShipCase5.txt

Begin

# Visible Manifest Name

Please Select Containers to Unload

UNUSED											
UNUSED											
UNUSED											
UNUSED											
UNUSED											
UNUSED											
UNUSED											
NAN	Cat	Dog	Pig	Hen	Rat	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED	NAN

# UI CONSIDERATIONS



**ENTER LOG NOTE**

RESTART

[LOG OUT](#)

**Current File:** SilverQueen.txt

**Job:** Load/Unload

## Begin

## Please Select Containers to Unload

# UI CONSIDERATIONS

The screenshot shows a user interface for a shipping application. At the top right, there is a language selection (EN) and a blue "LOG OUT" button. A large red arrow points from the bottom left towards the top right, labeled "Multiple User Options". Another blue arrow points from the top right towards the center, labeled "Log-Out Button".

**Loading Process**

Max Load Amount: 93

Containers To Load: 0

Enter Container Nam

Enter Container We

Add Container

Begin Unloading>Loading

If you do not have any containers to load, you may begin without adding containers.

# UI CONSIDERATIONS

The screenshot shows a mobile application interface for managing shipping containers. At the top, there is a header bar with a logo, 'ENTER LOG NOTE', 'RESTART', and 'LOG OUT' buttons. Below the header, the title 'Loading Process' is displayed. A red arrow points from the text 'List View of Containers to Load' to a list of three containers: 'cont 1 - 2kg', 'cont 2 - 4kg', and 'cont 3 - 6kg'. Each container entry includes a 'Delete' button. Another red arrow points from the text 'Option to Delete' to one of these delete buttons. At the bottom, there is a large blue button labeled 'Begin Unloading>Loading' and a note: 'If you do not have any containers to load, you may begin without adding containers.'

ENTER LOG NOTE   RESTART   LOG OUT

## Loading Process

Max Load Amount: 93

Containers To Load: 3

cont 1 - 2kg Delete

cont 2 - 4kg Delete

cont 3 - 6kg Delete

List View of Containers to Load

Add Container

Begin Unloading>Loading

If you do not have any containers to load, you may begin without adding containers.

Option to Delete

# MAINTENANCE PLAN

- Constraint updates
  - Ship capacity
  - Buffer capacity
  - Departure rules
  - Balance regulations
  - Occurrence:
    - Free: By call, for 5 years. Max once every 2 weeks.
- Program status
  - Are log files being generated correctly?
  - Are manifests being generated correctly?
  - New edge case?
  - Occurrence:
    - Free: Weekly for the first Quarter, then monthly for 1 year, then biyearly for 5 years.

# MAINTENANCE PLAN

- Optimizations
  - Performance monitoring
  - Algorithm improvements
  - Occurrence:
    - Free: Weekly for the first Quarter, then Quarterly for 1 year, then yearly for 5 years.
- Reports
  - Maintenance reports
  - Occurrence:
    - Free: Annually, for 6 years.
- General Support
  - Phone
  - Email
  - Occurrence:
    - Free 6 years.

# ACCEPTANCE TESTING: TIME

Test Case	Use Case	Testing Procedure	Validation
1	Testing Daylight Saving Time	<ul style="list-style-type: none"><li>- Use a timezone-aware library (moment-timezone in JavaScript)</li><li>- Convert all logs to PST by default and adjust for PDT only at the specific times of transition (moment-timezone in JavaScript)</li></ul>	<ul style="list-style-type: none"><li>- Set the server time to one hour before the PDT transition</li><li>- Confirm that timestamps log as YYYY-MM-DD 02:00</li></ul>
2	Testing time change from PDT to PST	<ul style="list-style-type: none"><li>- Use a timezone-aware library (moment-timezone in JavaScript)</li><li>- Convert all logs to PST by default and adjust back from PDT at transition time</li></ul>	<ul style="list-style-type: none"><li>- Server time will record 01:00 twice when falling back to PST from PDT</li><li>- Confirm that timestamps log as YYYY-MM-DD 01:00</li></ul>
3	Testing the server's system time with the Network Time Protocol time	<ul style="list-style-type: none"><li>- implement a checksum-based log verification mechanism</li></ul>	<ul style="list-style-type: none"><li>- Set up alerts to notify when server time deviates by a specified time (+/- 2 min)</li></ul>

# ACCEPTANCE TESTING: GENERAL

- We will have a final deliverable on or before December 13th, 2024
- We propose the following tests:
  - Two weeks before the acceptance test, you will send us up five scenarios (5 manifests and 5 corresponding transfer lists) and we will test any two of them you choose, live.
- The following are metrics of success.
  - Steps generation will take no longer than 15 minutes.
  - An operator will be able to logout, and another login and resume the job.
  - The steps provided will be optimal or near optimal. If other software or a human can provide steps that are significantly more optimal then the deadline & contract can be renegotiated or the project terminated.
  - The program must be able to resume when unexpectedly closed due to power failure or system crash.
  - Manifests and logs are correctly formatted
  - The software does not crash during testing.
  - The software can verify and maintain the correct time

# COMPLIANCE WITH REGULATIONS

## Automated Compliance with Customs Regulations

- Keeps logs of container details for customs inspections and comply with relevant national/international trade documentation requirements
- Based on the optimal balance plan or SIFT result, the program generates a step-by-step list of moves. Containers are reorganized to be legally balanced. When SIFT is used, it is logged.
- The system logs all actions for compliance audits and regulatory oversight
- Log file is not directly editable.

# **TRAINING AND DOCUMENTATION**

- We will provide a 10-15 minute video showing how to use the software.
  - What each button does.
  - What a load/unload job will look like.
  - What a balance job will look like.
  - Possible errors you may encounter.
  - Software setup.
- We will provide short documentation for usage.
  - Training will be better through the video.
- We will be available for questions via email.

# CONTRACT



- We will create software to solve the described task.
- We will have a final deliverable on or before December 6th, 2024.
- We may require up to 5 hours of your time to answer additional questions. Additional questions should be answered within 48 hours.
- We will not honor any requests for features not in this pitch, unless price and delivery date are renegotiated.
- After free support ends, future support can be provided regularly at a negotiated rate or in steeper one-off payments as needed.
- We retain the right to modify & sell the software to other companies
  - If and only if they are not located in Long Beach
  - Mr Keogh will receive a 7% royalty per sale, but not for support payments.

Signature (for KawrgoJumper): \_\_\_\_\_ Date \_\_\_\_\_

Signature (for Mr Keogh): \_\_\_\_\_ Date \_\_\_\_\_

# REFERENCES (General Understanding)

- [a] - Slide 15, Problem\_overview\_by\_Mr\_Keogh.pptx
- [b] - 1:05:46 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [c] -Slide 40, Problem\_overview\_by\_Mr\_Keogh.pptx
- [d] - Slide 44, Problem\_overview\_by\_Mr\_Keogh.pptx
- [e] - Slide 42 & 43, Problem\_overview\_by\_Mr\_Keogh.pptx
- [f] - Slide 43, Problem\_overview\_by\_Mr\_Keogh.pptx
- [g] - Slide 33, Problem\_overview\_by\_Mr\_Keogh.pptx
- [h] - Slide 33, Problem\_overview\_by\_Mr\_Keogh.pptx
- [i] - Slide 33, Problem\_overview\_by\_Mr\_Keogh.pptx
- [j] - 25:25 time mark, Elicitation Interview with Mr Keogh on 10/20/2024

# **REFERENCES (General Understanding)**

[k] - 25:51 time mark, Elicitation Interview with Mr Keogh on 10/20/2024

# +

# REFERENCES (Loading Unloading)

# +

[a] - Slide 23, Problem\_overview\_by\_Mr\_Keogh.pptx

[b] - Scenario described in email from Mr Keogh 10/22/2024, 11:49 AM

[c] - Scenario described in email from Mr Keogh 10/22/2024, 11:49 AM

[d] - Scenario described in email from Mr Keogh 10/22/2024, 11:49 AM

[e] - Slide 44, Problem\_overview\_by\_Mr\_Keogh.pptx



## **REFERENCES (Detailed Workflow)**

[a] - 15:45 time mark, Elicitation Interview with Mr Keogh on 10/20/2024

## REFERENCES (Assumptions)

- [a] - 43:27 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [b] - 13:55 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [c] - 21:04 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [d] - 50:05 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [e] - Slide 44, Problem\_overview\_by\_Mr\_Keogh.pptx
- [f] - 1:12:00 time mark, Elicitation Interview with Mr Keogh on 10/20/2024
- [g] - 1:01:50 time mark, Elicitation Interview with Mr Keogh on 10/20/2024

## **REFERENCES (Inputs)**

[a] - Elicitation Interview slide 18

[b] - Elicitation Interview slide 18

# THANKS!

Do you have any questions?

