

Project Pitch for Keogh Shipping



- Pitch by Tile Shipping Express
- Team: William Dang, Sarah Pun, Byron Bhuiyan, Liam Pinson, Javier Vargas
- Oct 29 2024

Our Understanding

- We provide two services: loading/unloading containers, and balancing containers. [1]
- Loading/unloading and balancing containers are two separate services and will never be done at the same time.
- For both services, our software should provide the fastest order of movements to complete the operation. [2]
- Our software will take the burden off your employee of having to manually come up with a plan and will provide the optimal solution. [2]
- During loading/unloading, ship balance does not matter so we focus on movements that lead to shortest time spent. [3]
- Any changes in crate placement must be reflected in an updated manifest. [4]
- The captain of the ship cannot legally leave until they receive an updated manifest. [4]
- We can approximate container movements to 1 minute per cell. [5]
- Movements between the ship and buffer are 4 minutes. [8]
- Movements between a truck and the ship or buffer are 2 minutes. [10]
- For balancing, a ship is considered balanced if the total mass on the port side and starboard side are within 10% of each other. [6]
- Our software will not interact with the transfer list in any way. [7]
- A log recording all actions (including sign in/out) should be kept. [9]

Assumptions

- The employees at Keogh's Shipping are able-bodied, in good shape, have perfect vision and fluent in English
- Workers will follow the program's instruction exactly, minimizing errors in loading/unloading
- Operators are only responsible for container placement, not container content
- All containers are the same size (40ft)
- When loading or unloading, balance is not considered
- The system includes a simple interface with two buttons: "Load/Unload" and "Balance"
- A visual grid with source and target locations is used, color-coded to make container placement straightforward
- No sign-outs events are logged; only sign-ins are recorded
- Backup generators ensure system recovery with a notification and data restoration
- Missing containers are handled by external parties
- Truck order is not prioritized, as worker bias is avoided by choosing optimally

Stakeholders

- Port Operators/Crane Operators
- Truck Drivers
- Mr. Keogh's Shipping Company
- Customers/Clients (the businesses that own the containers themselves)
- Supply chain Managers
- Head Office/Management (whoever deals with the logs and reports)
- Regulatory Agencies
- Maintenance Staff
- Insurance Companies
- IT/Technical Support
- Legal Teams
- Shipping Companies' Competition
- Local Government/City Planners
- Environmental Agencies
- Port Customers (Shippers/Freight Forwarders)

Logging in

- Starting the program just requires opening a web browser like Chrome and entering a web address

The diagram illustrates a simple login interface. At the top, there is a title bar with a wavy line icon and an 'X' on the left, and a minus, maximize, and close icon on the right. The main content area has the word 'Login' centered at the top. Below it, the label 'Name' is positioned to the left of a rectangular input field. Directly beneath the input field is a rectangular button labeled 'Login'.

- Upon opening the program, the employee is met with a login page that requires their name
- No password is required as the shipyard is guarded
- Upon signing in, this action is reflected in the log and the employee is taken to the dashboard

Dashboard

 X

  X

Tile Shipping Express

Welcome John

Log out

Upload Manifest

Choose File

 No file chosen

Select a Service

☐ Transfer ☐ Balance

View Log

Dashboard (corrected)

 X

— TB X

Tile Shipping Express

Welcome John

Log out

Select a Service

☐ Transfer ☐ Balance

View Log

Dashboard

- Upon logging in, the employee is shown the dashboard
- This screen has a welcome message, a prompt to upload the manifest, a prompt to select either transferring containers (loading/unloading) or balancing, and 3 buttons: one at the center of the screen to submit operation, one in the bottom right corner to view the log and the other in the top right to log out
- ~~● First, the employee is required to upload the manifest~~
- ~~● Next, they are required to select which service they want, whether that is transferring containers or balancing~~
- On this screen, the operator is meant to select which service they want, whether that is transferring or balancing containers
- ~~● After this, they press the submit button to perform the operations~~
- ~~The name of the ship and~~ the chosen operation will be reflected in the log

Manifest (continued)

~~~~~ X

— □ X

Upload Manifest

Upload

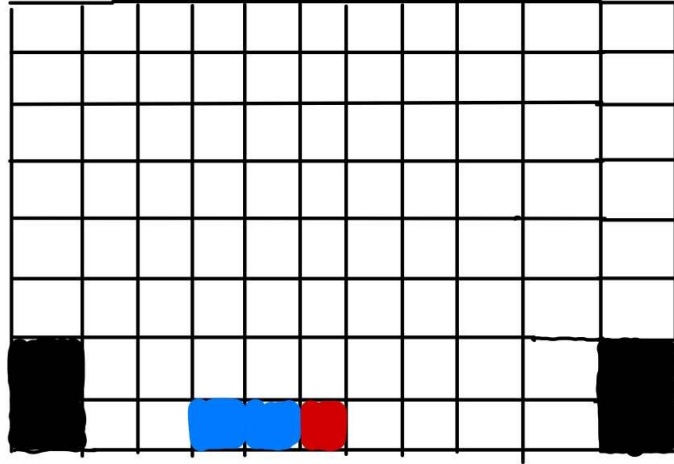
# Manifest

- On the next screen, the operator must upload the manifest
- After doing this, the ship's name will be reflected in the log
- The next screen will display the transfer or balance screen, depending on what the operator selected

## Transfer (Unloading)

## Unloading

First, select which containers should be unloaded



## Operations

Comments:

|  |
|--|
|  |
|--|

Next

Submit

Load

# Unloading

- The operator is met with a screen titled Unloading, with a 12 x 8 grid representing the ship and an empty box that says operations
- Our program processes the manifest and displays it in a user friendly manner
- Open spaces are white, unusable spaces are black and occupied spaces are blue
- Each space occupied by a container can be hovered over with the mouse that will display a text box with the container's name
- Unusable spaces are unable to be selected and will display nothing when hovered over
- If the operator selects a container to be removed, it will be highlighted in red
- If the operator changes his mind, they can simply click on it again to deselect it
- In this example, there are 3 containers in the ship, and the right most container is selected to be removed
- Lastly, there is a comment box that sends comments to the log after a comment is typed and enter is pressed

# Unloading (continued)

- If the operator wishes to unload nothing, they can just press “Load” and move on to the screen that allows them to load containers
- Assuming the operator wants to unload at least one container, they select which container(s) they want to unload and they press submit
- ~~● The operations box will display what steps are taken to move each container and the total amount of time this operation will take~~
- ~~● Our program will perform the solution and order that takes the least amount of time possible~~
- ~~● Each container movement will be shown one by one~~
- ~~● The fastest path will be highlighted for the operator to carry the container~~
- ~~● Assuming there is at least one more container to be moved, the operator must press next to move on to the next container~~
- ~~● If there is not and the button is pressed, the page will display an error saying there are no more operations to be performed~~
- ~~● Finally after unloading, the ship's manifest will be modified and the appropriate actions will be reflected in the log~~

# Transfer (Loading)

How many containers would you like to load?

Enter ~

# Transfer (Loading)

- At this point, the operator is in the loading phase
- After the unloading screen, the operator is met with a screen that asks how many containers they would like to load
- If they wish to load nothing, they can just enter 0 and skip all remaining screens to the end screen
- If the operator enters a positive number, a check will be performed to see if that number is less than or equal to the number of free spaces
- If so, they will move on to the next screen that allows them to enter the names of each container
- If the number exceeds the number of unoccupied spaces, an error will be thrown out telling the operator that there are only x number of available slots
- The operator will then be able to revise the number of containers to be loaded
- If the input is negative, the number of containers will be assumed to be 0

# Transfer (Loading) continued

- If the input is otherwise not a number, an error will be thrown out saying invalid input: must be a number
- The operations box will display what steps are taken to move each container and the total amount of time this operation will take
- Our program will perform the solution and order that takes the least amount of time possible
- Each container movement will be shown one by one
- The fastest path will be highlighted for the operator to carry the container
- Assuming there is at least one more container to be moved, the operator must press next to move on to the next container
- If there is not and the button is pressed, the page will display an error saying there are no more operations to be performed
- Finally after unloading, the ship's manifest will be modified and the appropriate actions will be reflected in the log



# Transfer (Loading)

Transfer

|  |  |  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Name

submit

Weight

submit

Comments:

Operations

Start

Next

# Transfer (Loading)

- For this example, let's say there are 3 containers to load
- The operator will be displayed a screen similar to the unloading screen with the ship grid and operations box as before but there are prompts to enter the name and weight of a container
- Pressing start on the bottom right will display the fastest path to unload the first container; after pressing this the button will disappear
- After the operator moves the container to the right spot, they must enter the name and weight of that container
- For the name, each input will be a string taking in any ASCII character
- If the operator tries to enter an input such as NAN or UNUSED, then there will be an error message for invalid input
- If the operator fails to provide a name for a container, ~~the name shown in the manifest will be displayed as "EMPTY"~~ there will be an error message for invalid input
- Upon finishing naming each container, the operator must input the weight of the container they had just lifted

# Transfer (Loading) continued

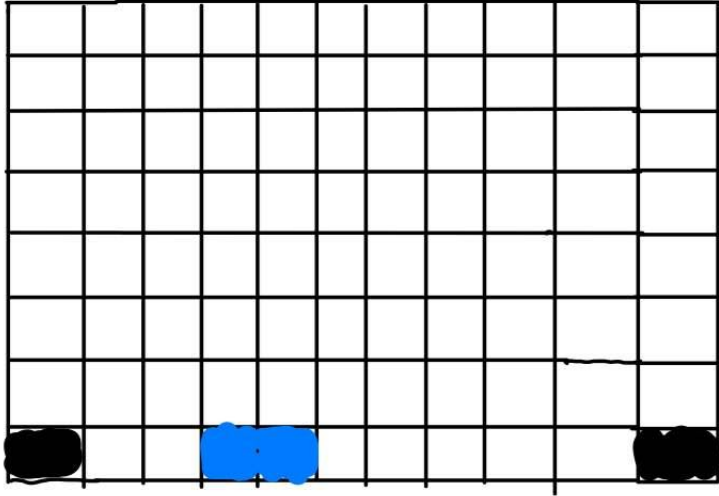
- This input must be a number from 0 to 99999
- If a negative number is inputted, that container will have a weight of 0 in the manifest
- If the weight exceeds 99999, that container's weight will be capped at 99999 in the manifest
- If any string that is not a number is inputted for the weight, an error for invalid input will be thrown out
- After entering the name and weight of a container, the operator will press "Next" and will be shown where to load the next container
- The operator will repeat the same naming process until every container is loaded
- Each container that is loaded will be reflected in the final manifest with its name, weight, and position

# Transfer (Loading) continued

- Similarly that to the unloading page, each operation and the amount of time it takes will be reflected in the operations box and each appropriate action will be reflected in the log
- When all containers are loaded, the operator will be guided to an end screen that says “Transfer Complete”
- ~~There will also be a button that allows them to~~ The program will automatically download the modified manifest which will be named <ShipsName>OUTBOUND.txt
- Finally there will be another button that allows the operator to return to the dashboard

# Balancing

Balance



## Operations

Comments:

|  |
|--|
|  |
|--|

Next

# Balancing

- If the operator wishes to balance the ship instead of loading/unloading, they press the balance button and submit instead
- They are taken to a screen titled “Balance” with the ship grid, operations and comment boxes, and a button that says “Next”
- Each container can be hovered over to see its name and weight
- The operator must click Next to see how to move the first container
- Initially, our program checks if the current ship can be balanced, if not, then all containers are unloaded to perform the SIFT operation
- If the ship can be balanced normally, each container and its movement is shown step by step by pressing next until the ship is moved
- Similarly to the transfer process, the operations box shows each operation and the amount of time it takes to move
- In the case that the SIFT operation is required, ~~each container will be moved off the ship to the buffer and~~ will be **sorted** heaviest at the center first, then so forth

# Balancing

- In this case, pressing next will also show each movement step by step
- After moving each container to its designated spot, the operator will be directed to the end screen
- Similarly to the transfer end screen, there will also be a button that allows them to download the modified manifest which will be named <ShipsName>OUTBOUND.txt
- There will be another button that allows the operator to return to the dashboard
- As such, every appropriate action will be reflected in the log

# Logging

```
2024-10-18 16:23 Joe has signed in.  
2024-10-18 16:23 QueenMary.txt was uploaded to the system.  
2024-10-18 16:23 Transfer was selected by operator.  
2024-10-18 16:23 QueenMary.txt was uploaded to the system.  
2024-10-18 16:23 Transfer was selected by operator.  
2024-10-18 16:25 QueenMary.txt was uploaded to the system.  
2024-10-18 16:25 Balance was selected by operator.  
2024-10-18 16:25 Balance is achievable.  
2024-10-18 16:25 Estimated Total Time for Service: 4 minutes  
2024-10-18 16:25 Container moved from: [1, 3] to [1, 7]  
2024-10-18 16:25 Balance service completed. Updated Manifest saved to manifests\QueenMaryOUTBOUND.txt
```



# Logging

- Hour and minutes (in PST) will be displayed
- Seconds are truncated so it is rounded down automatically
- This shows signing in and every operation that is done including the operation selected, and the movement and name of each container moved
- In the top right corner there will be a button that allows the operator to download the log
- The file will be named KeoghsPort2024.txt (since it is 2024)
- The file will be named accordingly to the current year

# Inputs

- Container information
  - Name
  - Weight
- Manifest
  - Software will automatically read the manifest to create and display grid
- Comments
  - Employee can submit comments when necessary during either service.
  - Comments will be displayed in the action log.
- Employee interaction
  - Employee will manually click on-screen prompts to advance the operation

# Output

- Updated manifest
  - Automatically made once either service is complete.
  - Will have “OUTBOUND” appended to name to signify service completion.
- Action log
  - Records all actions with timestamps

# Scenario I: Part 1 of 10

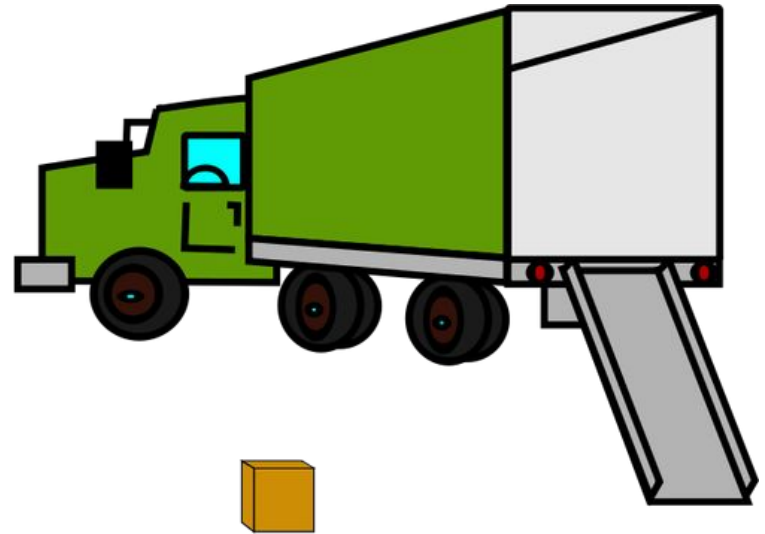
- John is a 3 year employee of Keogh Shipping.
- He speaks fluent English and has a highschool diploma.
- It is 11:30am.
- John has been working since 4am and is looking forward to clocking out at 12pm.
- Currently there are no ships docked at the port.

# Scenario I: Part 2 of 10

- At 11:32, John is sitting in the cabin waiting when we receives a transfer list.
- He begins reading through it to know what he will have to do for the incoming ship.
- John reads the following: *The Blue Grand will come in today at about 12:00pm. Please load Ty's Tyres, Autozone Windshield Wipers, and Staples Notebooks, and unload Walmart NC Car Toys. All drivers have been notified and are on their way.*
- In total, John has to load 4 containers onto the ship and unload 1 container from the ship.

# Scenario I: Part 3 of 10

- At 11:40, John sees four trucks loaded with containers pull into the waiting area. These are likely the four trucks with the containers he needs to load onto the ship.
- Shortly after a truck with no container pulls in. This is likely for the container he has to unload from the ship.



# Scenario I: Part 4 of 10

- At 11:45, John receives an email from the captain of the Blue Grand with the manifest.
- The manifest is titled `BlueGrand.txt`
- John notices that the BlueGrand is almost at full capacity with only 2 unused slots.
- He realizes he will not be able to load all 4 containers onto the ship like the transfer list specified.

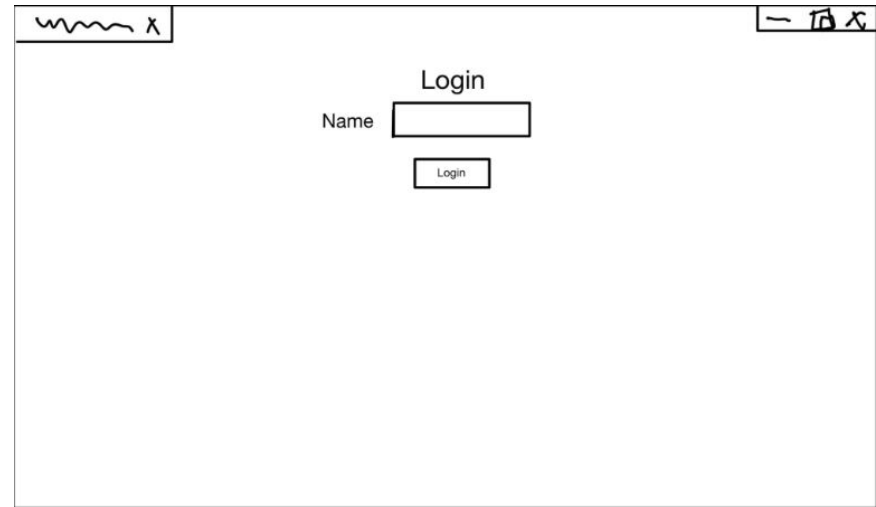
BlueGrand.txt - Notepad

File Edit Format View Help

[01,01], {00388}, Strong's Tools  
[01,01], {00388}, Strong's Tools  
[01,01], {00388}, Strong's Tools  
::::::::::::::::::::::::::::::::::::  
[07,01], {00000}, UNUSED  
[07,02], {01243}, Von's Cutlery  
::::::::::::::::::::::::::::::::::::  
[08,01], {00000}, UNUSED  
[08,02], {00874}, Walmart NC Car Toys  
::::::::::::::::::::::::::::::::::::

# Scenario I: Part 5 of 10

- At 12:00, Ben taps John on the shoulder, signaling the end of John's shift.
- Ben is another employee who has worked for Keogh Shipping for the past 8 months.
- When John exits the cabin, Ben sits down and signs in.
- He does this by clicking on the logout button which takes him to the main menu of the program.
- Here he will enter his name and log in.
- Upon doing this, an entry is entered in the log file stating that *John has signed out* and that *Ben has signed in*.



Wavy line X

— □ X

Login

Name

Login



# Scenario I: Part 6 of 10

- At 12:01, Ben begins reviewing the transfer list and manifest to get up to speed.
- The Blue Grand pulls into the port shortly after.
- Ben starts with the container that needs to be unloaded and makes an announcement to the truck waiting area.
- Ben says “*Truck for Walmart NC Car Toys please pull into the loading area*”.
- Ben then begins the process unloading process.



# Scenario I: Part 7 of 10

- To begin, Ben clicks on the *Choose File* button and uploads the manifest of the Blue Grand.
- The manifest will be used to generate a grid for Ben to interact with when loading and unloading containers.
- After uploading the manifest, Ben clicks on the *Transfer* button.
- Here Ben further specifies whether he will be *Loading* or *Unloading*.
- In this case, Ben is unloading so he clicks on *Unload*.

Upload Manifest

Choose File

No file chosen

1. Load

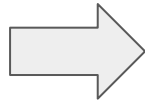
2. Unload

# Scenario I: Part 8 of 10

- Upon clicking *Unload*, Ben clicks on the cell with the container to be unloaded.
- The cell will turn green to indicate that it has been selected.
- When Ben is done clicking all cells that need to be unloaded, in this case just 1, Ben will confirm he is done selecting and ready to begin physically moving the containers.
- When Ben confirms, the program will automatically compute the optimal order of operations that will result in the fastest unload.
- Once computed, the cell that needs to be moved first will be highlighted, indicating to Ben that he begins the unloading process by moving the crate at the highlighted cell.

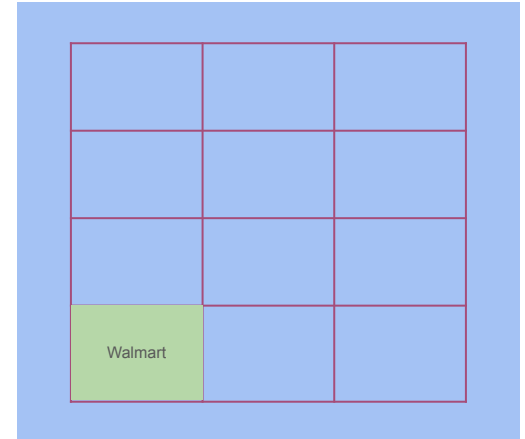
1. Loading/Unloading

2. Balance



1. Load

2. Unload



# Scenario I: Part 9 of 10

- Once Ben has gone through all of the recommended operations to unload the container, a prompt will appear informing him that the container has been unloaded.
- The truck, now with the container, drives off and Ben makes the next announcement.
- “*Truck for Ty’s Tyres please move to loading area.*”
- In the same screen Ben is in, he now clicks *Load* instead of *Unload*.
- Once he clicks *Load*, Ben will see a prompt asking for the container name and weight.

New Container

Name:

Weight:

# Scenario I: Part 10 of 10

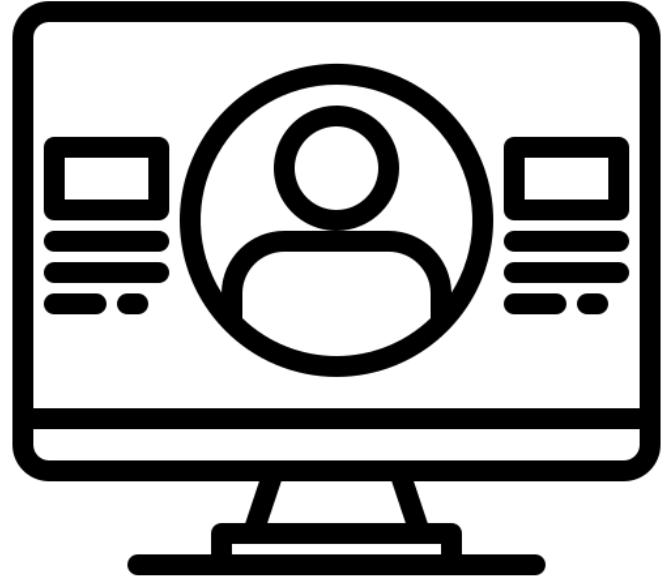
- At this point, there are only 3 unused slots aboard the Blue Grand, and 4 containers that need to be loaded.
- One will not be loaded.
- Ben begins by physically picking up the container from the truck.
- When he does this, the weight of the container is displayed to him and he can then enter the info into the prompt.
- Once entered, the optimal path for loading the container will appear on the screen and Ben will follow the instructions.
- Once he finishes this for the first container, the process will repeat for all remaining containers.
- However, upon loading the 3rd container, a prompt appears informing Ben that the ship is full!
- No more containers can be loaded.
- Ben announces this to the last truck and it drives off to come back another day.
- The manifest is updated with all relevant changes and sent back to the captain of the Blue Grand.
- Once received, the Blue Grand leaves the port.

# Scenario 2: Part 1 of 5

- Alex Johnson is a 15-year employee at the Port of Long Beach.
- He works full-time as the Dock Operations Manager.
- He has a bachelor's degree in logistics management.
- He manages a large team and oversees several container operations each week

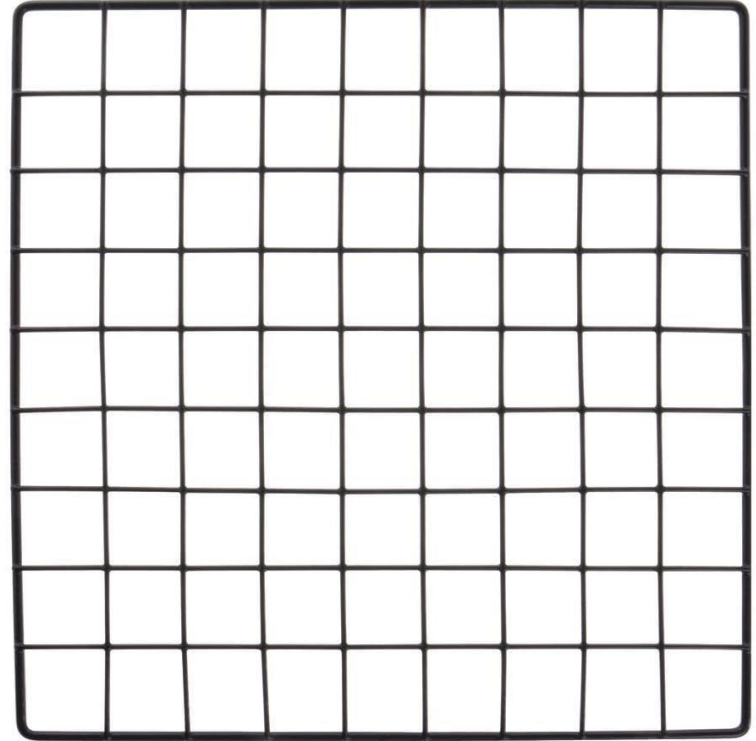
# Scenario 2: Part 2 of 5

- Alex Johnson wakes up at 3:00 AM on October 27th, 2024
- He arrives at the port of Long Beach at 4:00AM
- He clicks sign in so it logs him in
- He waves to the current crane operator to let them know that their shift is over
- He reviews the manifest for the ship, Ocean Star, ensuring everything is in order for the shift



## Scenario 2: Part 3 of 5

- The software has already generated the optimal loading plan, saving him the need to calculate it manually
- He checks to make sure the containers are ready to be moved
- There is a transfer button he clicks to get to the load/unload button





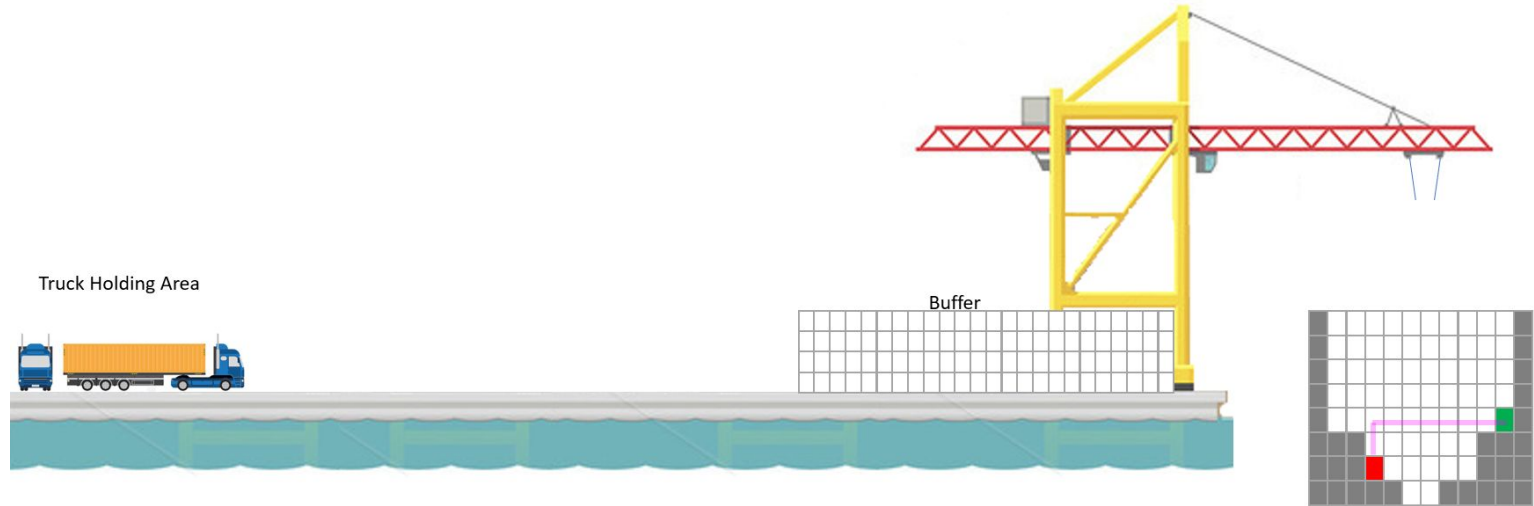
## Scenario 2: Part 4 of 5

- On the transfer list, it says an oncoming container with car tires from Honda
- He sees the truck containing car tires from Honda by the buffer zone.
- He clicks the load button and can see the container move from the buffer zone into the ship taking the most optimal time to load
- While it is moving, he types into the manifest 'HondaTires' at 4:39AM so that it logs what is going onto the ship



# Scenario 2: Part 5 of 5

- Alex moves the crane back to the buffer zone and waits for another truck to come and repeat the same steps



## Scenario 3: Part 1 of 7

- Bob is in the cabin.
- Bob gets an email from head office. The email states:
  - “The Zephyr will be there around 7:45, please load *Roberts Coolers*, I have called their truck driver”



## Scenario 3: Part 2 of 7

- The truck for *Roberts Coolers* pulls into the waiting area to be unloaded
- Bob receives another email from the Captain of the *Zephyr*, it comes with a manifest titled “ZephyrManif.txt”



## Scenario 3: Part 3 of 7

- The ship docks
- Bob broadcasts for the truck in the waiting area to pull forward to be unloaded



## Scenario 3: Part 4 of 7

- Bob selects load, but as the program is calculating the best spot to put the container, there is a power outage.
- Bob waits a couple of minutes for the Power Generator to kick in.



## Scenario 3: Part 5 of 7

- The system boots back up.
- The program resumes where it left off.
- It shows Bob the spot [02,01].



## Scenario 3: Part 6 of 7

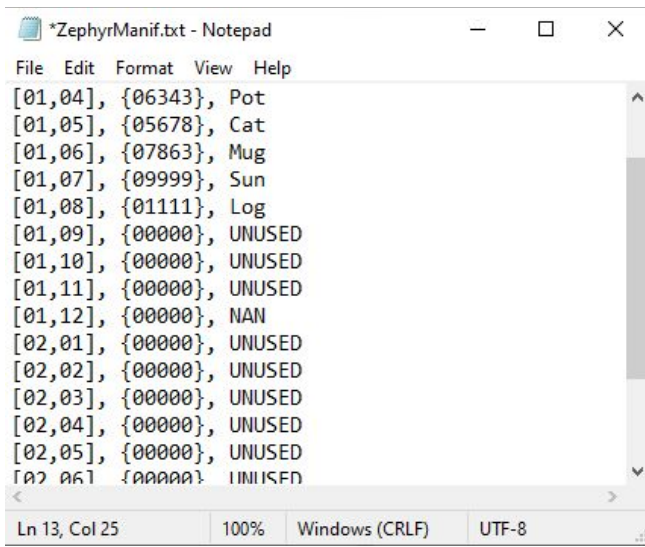
- Using the crane, Bob grabs the *Roberts Cooler* container
- He weights the container and writes down the weight, 1783 kilos.
- He loads it at the given location [02,01]





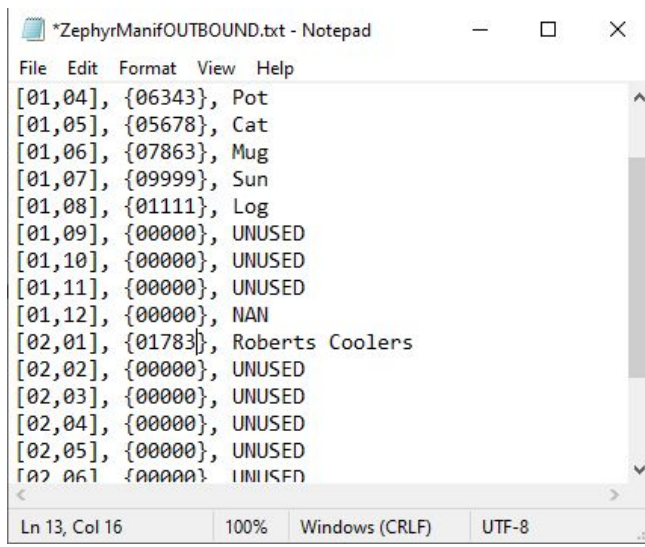
## Scenario 3: Part 7 of 7

- Bob opens the manifest, ZephyrManif.txt
- He edits the file to reflect the new load
- He renames the file to ZephyrManifOUTBOUND.txt and emails it to the captain
- The Ship departs



A screenshot of a Notepad window titled '\*ZephyrManif.txt - Notepad'. The window contains a list of items in a structured format. The status bar at the bottom shows 'Ln 13, Col 25', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
File Edit Format View Help
[01,04], {06343}, Pot
[01,05], {05678}, Cat
[01,06], {07863}, Mug
[01,07], {09999}, Sun
[01,08], {01111}, Log
[01,09], {00000}, UNUSED
[01,10], {00000}, UNUSED
[01,11], {00000}, UNUSED
[01,12], {00000}, NAN
[02,01], {00000}, UNUSED
[02,02], {00000}, UNUSED
[02,03], {00000}, UNUSED
[02,04], {00000}, UNUSED
[02,05], {00000}, UNUSED
[02,06], {00000}, UNUSED
```

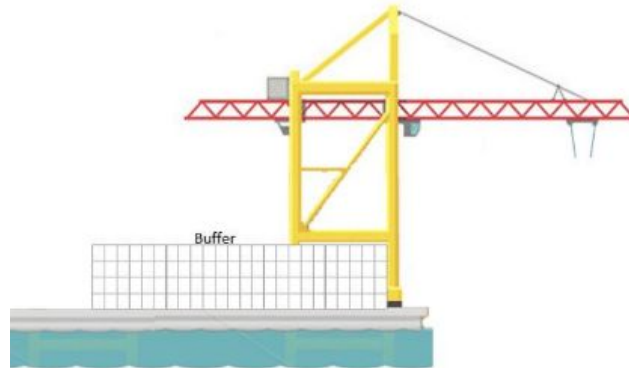


A screenshot of a Notepad window titled '\*ZephyrManifOUTBOUND.txt - Notepad'. The window contains the same list of items as the previous window, but with one modification: the entry for [02,01] is now '{01783}, Roberts Coolers' instead of 'UNUSED'. The status bar at the bottom shows 'Ln 13, Col 16', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
File Edit Format View Help
[01,04], {06343}, Pot
[01,05], {05678}, Cat
[01,06], {07863}, Mug
[01,07], {09999}, Sun
[01,08], {01111}, Log
[01,09], {00000}, UNUSED
[01,10], {00000}, UNUSED
[01,11], {00000}, UNUSED
[01,12], {00000}, NAN
[02,01], {01783}, Roberts Coolers
[02,02], {00000}, UNUSED
[02,03], {00000}, UNUSED
[02,04], {00000}, UNUSED
[02,05], {00000}, UNUSED
[02,06], {00000}, UNUSED
```

## Scenario 4: Part 1 of 9

- Dylan is a 2 year employee of Keogh shipping.
- He works full-time, Monday through Friday
- He has a high school diploma and is fluent in English.
- He arrives on site at around 8am to start his shift in the crane cabin.
- There is no ship on the dock and there are no trucks.

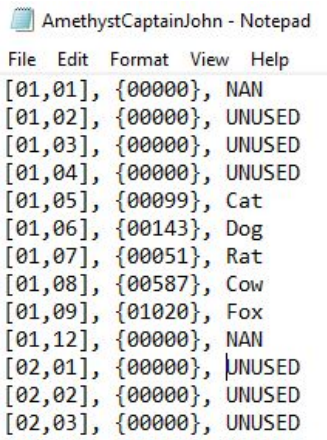


## Scenario 4: Part 2 of 9

- Dylan gets an email from the head office; it is a transfer list that reads:  
The Amethyst will dock at around 2pm. The captain requests us to balance the ship.

## Scenario 4: Part 3 of 9

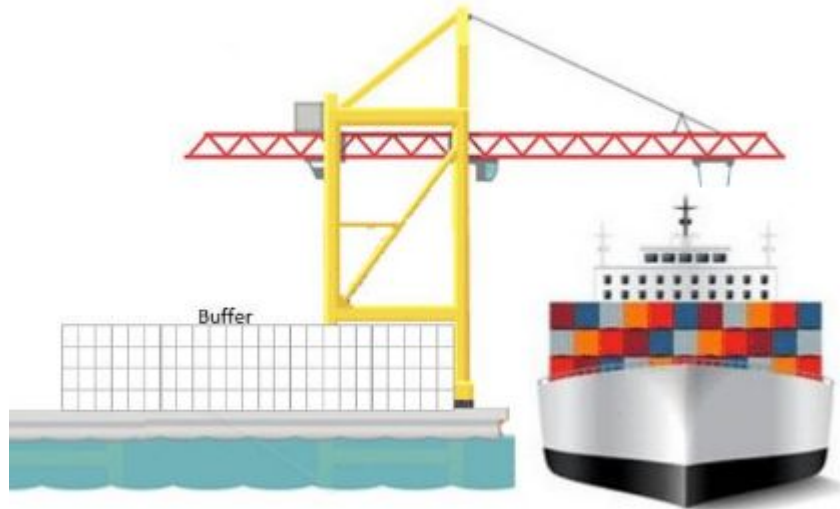
- The captain of the Amethyst sends an email the Dylan with the manifest “AmethystCaptainJohn.txt”
- Dylan looks at it and sees 5 different containers with very different weights that would make the ship unbalanced.



```
AmethystCaptainJohn - Notepad
File Edit Format View Help
[01,01], {00000}, NAN
[01,02], {00000}, UNUSED
[01,03], {00000}, UNUSED
[01,04], {00000}, UNUSED
[01,05], {00099}, Cat
[01,06], {00143}, Dog
[01,07], {00051}, Rat
[01,08], {00587}, Cow
[01,09], {01020}, Fox
[01,12], {00000}, NAN
[02,01], {00000}, UNUSED
[02,02], {00000}, UNUSED
[02,03], {00000}, UNUSED
```

## Scenario 4: Part 4 of 9

- At 2:07pm the ship docks

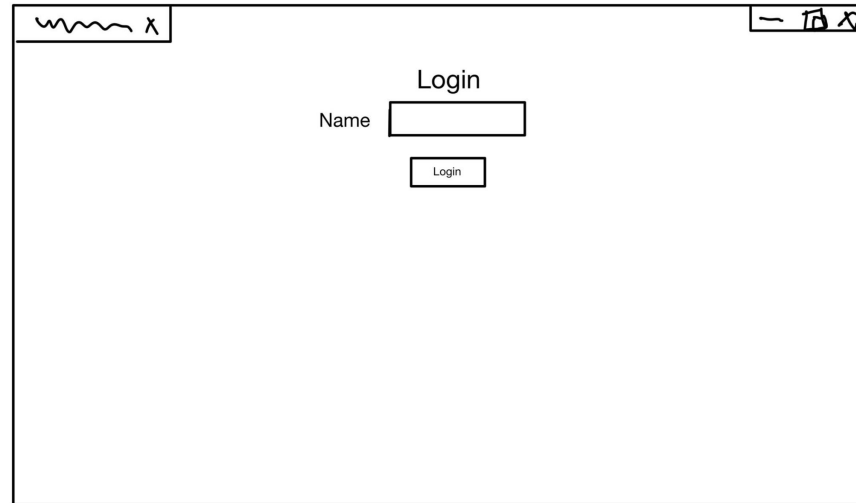


## Scenario 4: Part 5 of 9

- Dylan turns on the computer and launches the software provided by Tile Shipping Express.

## Scenario 4: Part 6 of 9

- Dylan logs into the system.
- Dylan provides the text file that the captain of the Amethyst sent into the software.

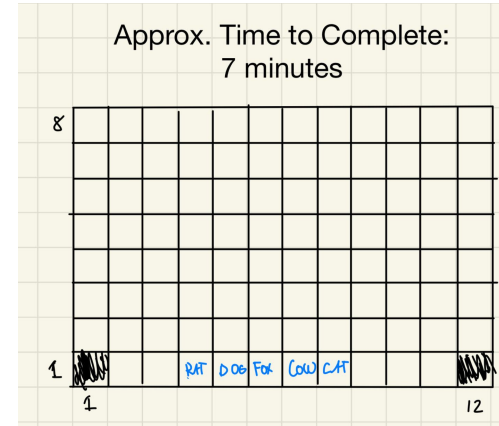


The image shows a simple login form interface. It has a title bar at the top with a wavy line icon on the left and a close button (X) on the right. The main area contains the word "Login" centered at the top. Below it, the word "Name" is followed by a rectangular input field. Below the input field is a rectangular button labeled "Login".

## Scenario 4: Part 7 of 9

- He then chooses the balancing option and is given a visual representation of where the containers will be on the ship grid and an approximate time for when the action will be completed.

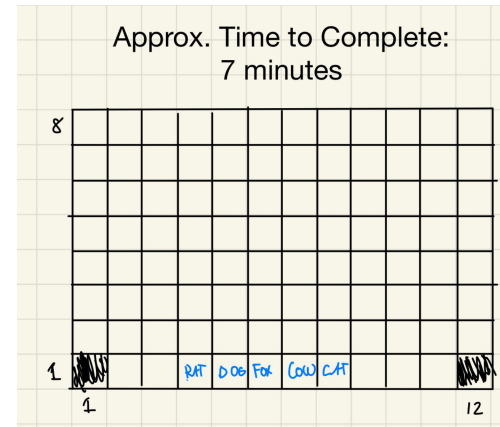
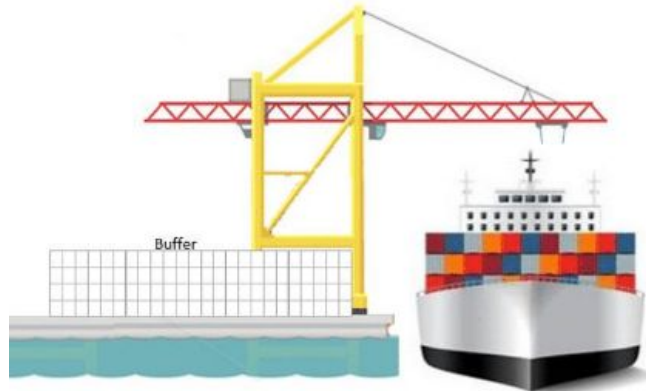
The screenshot shows a web browser window with the title 'Tile Shipping Express'. The page content includes a 'Welcome Dylan' message, an 'Upload Manifest' section with a 'Choose File' button and 'No file chosen' text, and a 'Select a Service' section with radio buttons for 'Transfer' and 'Balance', and a 'Submit' button. There are also 'Log Out' and 'View Log' buttons. The browser window has a wavy line icon and an 'X' in the top left corner, and a minus sign, 'TB', and 'X' in the top right corner.





## Scenario 4: Part 8 of 9

- Dylan proceeds with moving the containers to the correct locations on the ship, documenting each container's weight to make sure they align with the weight specified in the manifest.



## Scenario 4: Part 9 of 9

- After completing the balancing process, the software edits the manifest to reflect the changes and outputs a text file  
“AmethystCaptainJohnOUTBOUND.txt”
- Dylan emails this to the ship captain and the ship departs.

# Training and Documentation

- We offer complimentary user support and training for any changes made under our maintenance plan **such as video training or an in person training before the first start date.**
- Simple instruction manual available to operator at any time
  - Step by step instructions on how to operate and navigate through the programs functions
  - Important information highlighted for the operators to keep a note of.

# Compliance with Regulation

- The system will comply with all the usual contract/client regulations, namely the **California Consumer Privacy Act (CCPA)** and **Federal Trade Commission (FTC) Act**.
  - These acts protect your data and information and serve to prevent any misuse of it while our system is in use.
  - A comprehensive lists of all relevant regulations and compliances will be provided at a later date but before the final deliverable is handed over.

# Maintenance Plan

While we cannot anticipate every possible change, we recognize that certain developments may require software updates or adjustments in the years following delivery.

- If new regulations require modifications to ship balancing or loading protocols (e.g., stricter weight distribution laws or additional documentation), we will implement the necessary changes for free within the first three years of delivery.
- Should hardware requirements evolve (e.g., if the system needs to adapt to new crane models or automated trucks), we will make adjustments to ensure compatibility, free of charge, within three years of delivery.
- If operational expansions require the system to handle additional ships or higher cargo volumes, we will provide an expansion plan at a discounted rate within five years, covering scalability and performance adjustments.

# Acceptance Testing

- We will have a final deliverable on or before December 12<sup>th</sup> 2024 (or no more than 5 days after acceptance testing).
- We propose the following tests:
  - Two weeks before the acceptance test, you will send us up five scenarios (scripts and data files) and we will test any two of them you choose, live.
  - With zero notice, you will provide up to three scenarios, and we will test them live.
- The following are metrics of success:
  - The load/unload algorithm should take no more than 30 seconds to generate an optimized solution for which containers to move and in which order.
  - The balancing algorithm should take no more than 30 seconds to generate an optimized solution for which containers to move and in which order.
  - Each process should always provide a solution.
  - User should be able to login and load the software with the manifest.

# Intellectual Rights and Licensing

- Mr. Keogh will hold exclusive usage rights for two years during which we agree not to sell or license the software to any direct competitors in the port or logistics industry.
- After the exclusivity period, if our team decides to license or sell the software to other parties, including Mr. Keogh's competitors, we agree to a profit-sharing arrangement that provides Mr. Keogh 10% of any net profits from such sales.
- We commit to a non-disclosure agreement that restricts sharing any sensitive or proprietary information specific to Mr. Keogh's operations
- If Mr. Keogh prefers, we can negotiate a buyout option where he gains full ownership of the intellectual property, granting him the freedom to control future licensing or resale of the software.

# Contract

- We propose to create a system that will solve the task at hand
- We will have a final deliverable on or before December 12<sup>th</sup> 2024 (or no more than 5 days after acceptance testing).
- We may require up to five hours of your time (or the time of a qualified proxy) to answer any additional questions. Questions should be answered within 48 hours.
- We will not honor any requests for “feature creep”, at this price point and delivery date.

Signed (for Mr. Keogh’s Company) \_\_\_\_\_ Date \_\_\_\_\_

Signed (for Tile Shipping Express) \_\_\_\_\_ Date \_\_\_\_\_



# References

- [1] Problem Overview Presentation on Oct 7 2024, slide 14
- [2] Problem Overview Presentation on Oct 7 2024, slide 24
- [3] Problem Overview Presentation on Oct 7 2024, slide 36
- [4] Problem Overview Presentation on Oct 7 2024, slide 21
- [5] Problem Overview Presentation on Oct 7 2024, slide 41
- [6] Problem Overview Presentation on Oct 7 2024, slide 37
- [7] Problem Overview Presentation on Oct 7 2024, slide 23
- [8] Problem Overview Presentation on Oct 7 2024, slide 43
- [9] Problem Overview Presentation on Oct 7 2024, slide 46
- [10] Problem Overview Presentation on Oct 7 2024, slide 44