Take-Home\_Ex03 - Storyboard

Celin Liaw, Shermainn Ng, Stefanie Felicia

2025-06-11

# Introduction

This storyboard will contain the tasks and questions from [VAST Challenge](https://vast-challenge.github.io/2025/index.html) and [Mini Challenge 3](https://vast-challenge.github.io/2025/MC3.html). The visualisations will help to answer questions from MC3.

Our storyboard helps break down Oceanus’s complex story into clear, visual scenes that are easy to follow. It allows viewers to understand key events, characters, and conflicts without being overwhelmed by information.

The story begins with Oceanus shifting from a fishing-based economy to tourism after illegal fishing crackdowns. Tensions grow as former players reinvest and power dynamics shift, reaching a peak when pop star Sailor Shift arrives to film a music video.

Our focus is on investigative journalist Clepper Jessen, who uncovers suspicious activity behind the sudden closure of Nemo Reef. By tracking intercepted communications—radio chatter, messages, and secret meetings—Clepper pieces together a web of rushed permits, hidden deals, and corruption.

The storyboard will follow these communications closely, helping the audience connect the dots as the deeper story unfolds.

# Motivation

In order to find more detailed insights, communication visualizations as well as other types of visualizations will be used to help Clepper to uncover and confirm his suspicions based on the analysis that we will be doing to identify important events and relationships between the people, organizations and vessels in the 2 week period.

With the use of the storyboard, user would be able to visualize the analysis more effectively with the data set provided within the 2 weeks. Based on the communications, user can focus on the content as well as identify who are the suspicious players that play a part in the illegal fishing activities.

# Methodology

The storyboard aims to showcase my team’s ideas on how to show and map out the user’s experience.

The prototype can be broadly classify into 3 key areas:

* Heat Map Analysis
* Network Graph Communication Analysis
* Timeline Analysis

For enhanced user experience, the prototype will include filter components (i.e. parameter selections such as entities that we would want to select) as well as timeline slider bar that shows us the time period of events that the user would like to focus on. A legend is also shown to allow user to be identify clearly on the different types of nodes displayed on the plots.

# Storyboard

## 1. Importing Libraries

All these libraries will be used to build the Shiny website.

* shiny - Build and serve interactive web apps entirely in R
* jsonlite - Read, write and manipulate JSON data quickly
* dplyr - Filter, mutate and summarise
* purr - For elegant list/vector iteration
* visnetwork - create interactive network graphs
* DT - Render Interactive Data Tables
* tidygraph - Tidy-style verbs for graph data (nodes/edges) manipulation
* ggraph - graphics plotting for network/graph objects
* patchwork - combine ggplots
* grid - low level graphics system underpinning lattice, ggplot
* ggplot2 - build layered graphics

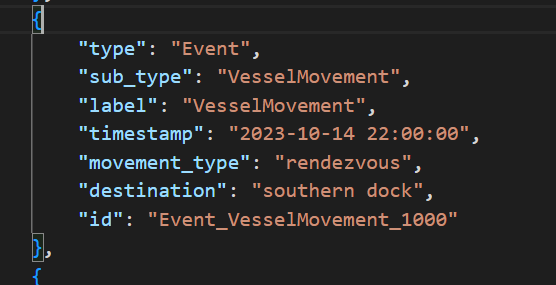
library(shiny)  
library(jsonlite)  
library(dplyr)  
library(purrr)  
library(visNetwork)  
library(DT)  
library(tidygraph)  
library(ggraph)  
library(patchwork)  
library(grid)  
library(ggplot2)

## 2. Creating Legend

Give a legend to the different nodes

legend\_cols <- c(  
 Person = "#1b9e77",  
 Vessel = "#d95f02",  
 Organization = "#7570b3",  
 Relationship = "#e7298a",  
 Group = "#e6ab02",  
 Location = "#66a61e"  
)

## 3. Load and Process Data

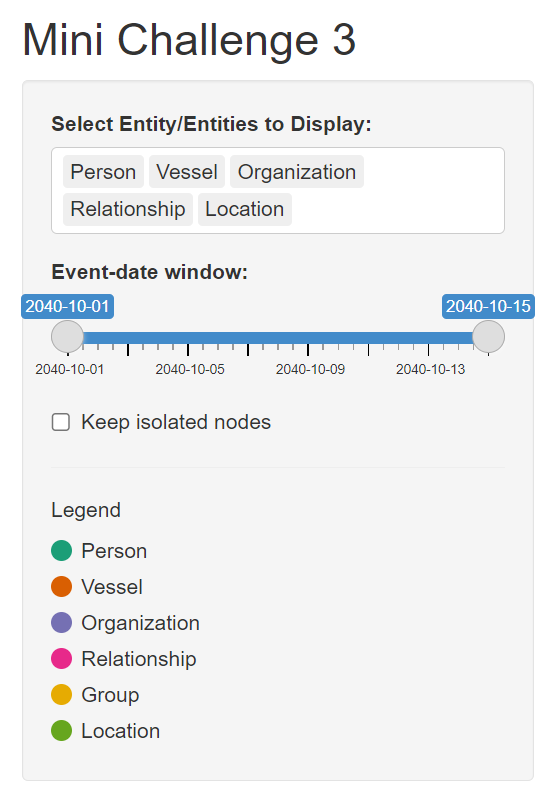
* **read\_json** - reads the json files
* **simplifyVector = TRUE** - ensure the nested arrays become data frames called graph$nodes and graph$edges
* **events\_df** - keeps only nodes where type == Event and timestamp is not missing
* **filter** - to keep only dates from 2040-10-01 to 2040-10-15 (this is due to an error in the data where there is a year 2023, while all other data are from year 2040)
* 
* **nodes\_all** - used to create a new column category where type == Relationship and to label as “Relationship” and to only keep nodes that are in the legend category
* **edges\_all** - rename source and target to “from” and “to” and adds a fixed “to” arrow direction for plotting the visnetwork
* A new column event\_date is also created

graph <- read\_json("Data/MC3\_graph.json", simplifyVector = TRUE)  
  
events\_df <- graph$nodes %>%  
 filter(type == "Event", !is.na(timestamp)) %>%  
 mutate(date = as.Date(timestamp)) %>%  
 filter(date >= as.Date("2040-10-01"), date <= as.Date("2040-10-15"))  
  
min\_date <- min(events\_df$date)  
max\_date <- max(events\_df$date)  
  
nodes\_all <- graph$nodes %>%  
 mutate(category = if\_else(type == "Relationship", "Relationship", sub\_type)) %>%  
 filter(category %in% names(legend\_cols)) %>%  
 mutate(color = legend\_cols[category])  
  
edges\_all <- graph$edges %>%  
 transmute(from = source,  
 to = target,  
 type = type, # 'sent', 'received', etc.  
 arrows = "to") %>%  
 left\_join(events\_df %>% select(id, date), by = c("from" = "id")) %>%  
 rename(date\_from = date) %>%  
 left\_join(events\_df %>% select(id, date), by = c("to" = "id")) %>%  
 rename(date\_to = date) %>%  
 mutate(event\_date = coalesce(date\_from, date\_to)) %>%  
 select(-date\_from, -date\_to)

## 4. UI Design

* **sidebarLayout** - Used to create the controls for the display on the right
* **selectInput** - Used to select which Entities to display
* **sliderInput** - Used to change the date of event
* **checkboxInput** - For removing isolated nodes
* **mainPanel** - Used to create the visualisation for the UI
* **tabPanel** - To create the different tabs to switch between visualisations

ui <- fluidPage(  
 titlePanel("MC3 Entity & Relationship Network"),  
 sidebarLayout(  
 sidebarPanel(  
 selectInput("classes", "Select Entity/Entities to Display:",  
 choices = names(legend\_cols),  
 selected = c("Person", "Vessel"),  
 multiple = TRUE),  
 sliderInput("dateRange", "Event-date window:",  
 min = min\_date, max = max\_date,  
 value = c(min\_date, max\_date),  
 timeFormat = "%Y-%m-%d"),  
 checkboxInput("isolates", "Keep isolated nodes", FALSE),  
 tags$hr(), tags$h5("Legend"),  
 ## dynamic legend based on legend\_cols -------------------------  
 tagList(lapply(names(legend\_cols), function(cat) {  
 tags$div(style="display:flex; align-items:center; margin-bottom:4px;",  
 tags$span(style=sprintf(  
 "display:inline-block;width:14px;height:14px;border-radius:50%%;\  
 background:%s;margin-right:6px;", legend\_cols[cat])),  
 tags$span(cat)  
 )  
 })),  
 width = 3  
 ),  
 mainPanel(  
 tabsetPanel(id = "tabs",  
   
 tabPanel("Data Table",  
 tabsetPanel(type = "tabs",  
 tabPanel("Nodes", DTOutput("nodes\_table\_dt")),  
 tabPanel("Edges", DTOutput("edges\_table\_dt"))  
 )  
 ),  
   
 tabPanel("Heatmap & Frequency of Communication",  
   
 selectInput("focus\_id", "Select by id",  
 choices = "All", selected = "All"),  
 visNetworkOutput("pv\_net", height = "700px")  
 ),  
   
 # visNetwork   
 tabPanel("Interactive Network",  
 visNetworkOutput("net", height = "800px"), br(),  
 tabsetPanel(type = "tabs",  
 tabPanel("Nodes", DTOutput("nodes\_table")),  
 tabPanel("Edges", DTOutput("edges\_table"))  
 )  
 ),  
   
 # original daily plot   
 tabPanel("Daily Communication Graph & Timeline",  
 plotOutput("dailyPlot")  
 )  
  
 )  
 )  
 )  
)



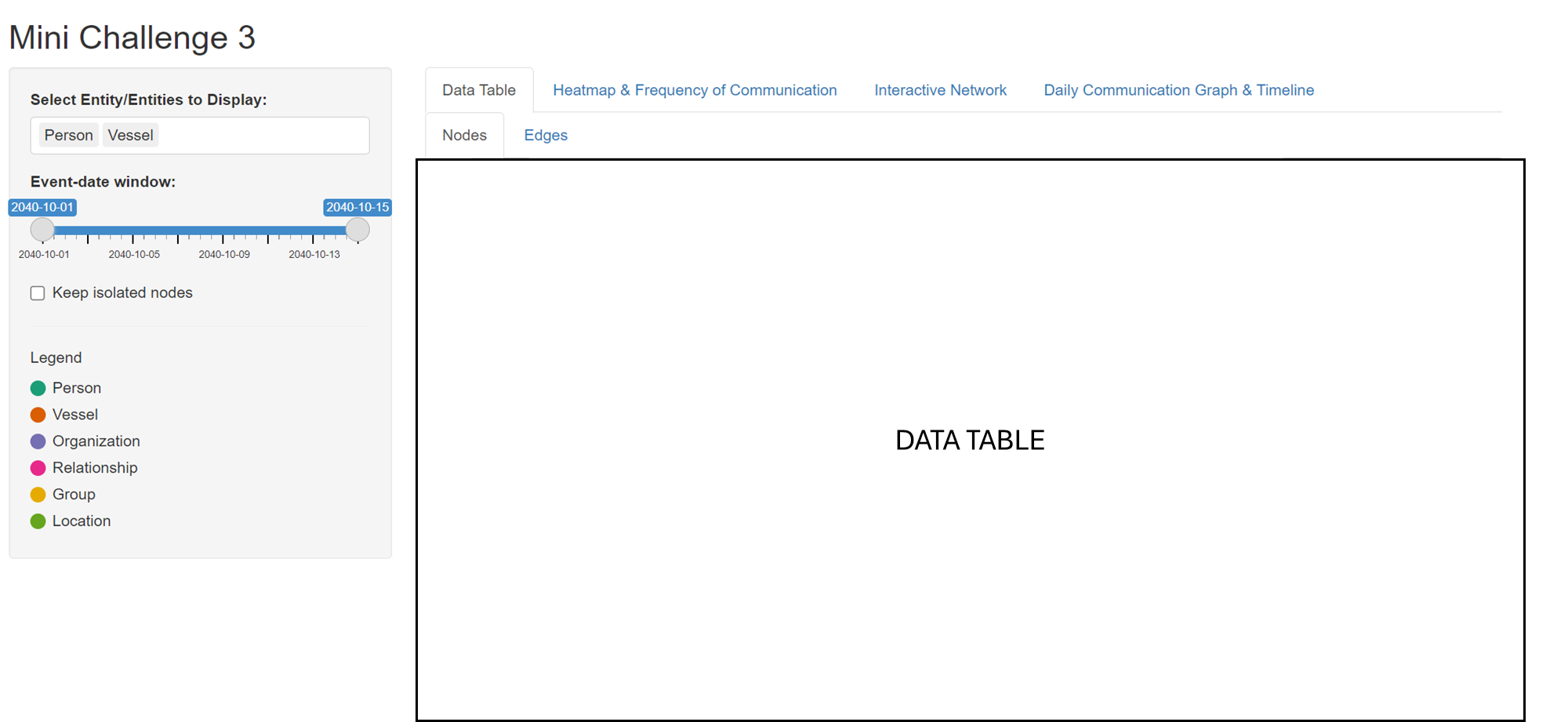
This sidebar layout enables the user to filter out specific entities and their dates so that user is able to zoom in and focus on the details of the interaction and relationship between the entities.

## 5. Final Visualisations

### 1st Tab - Data Table

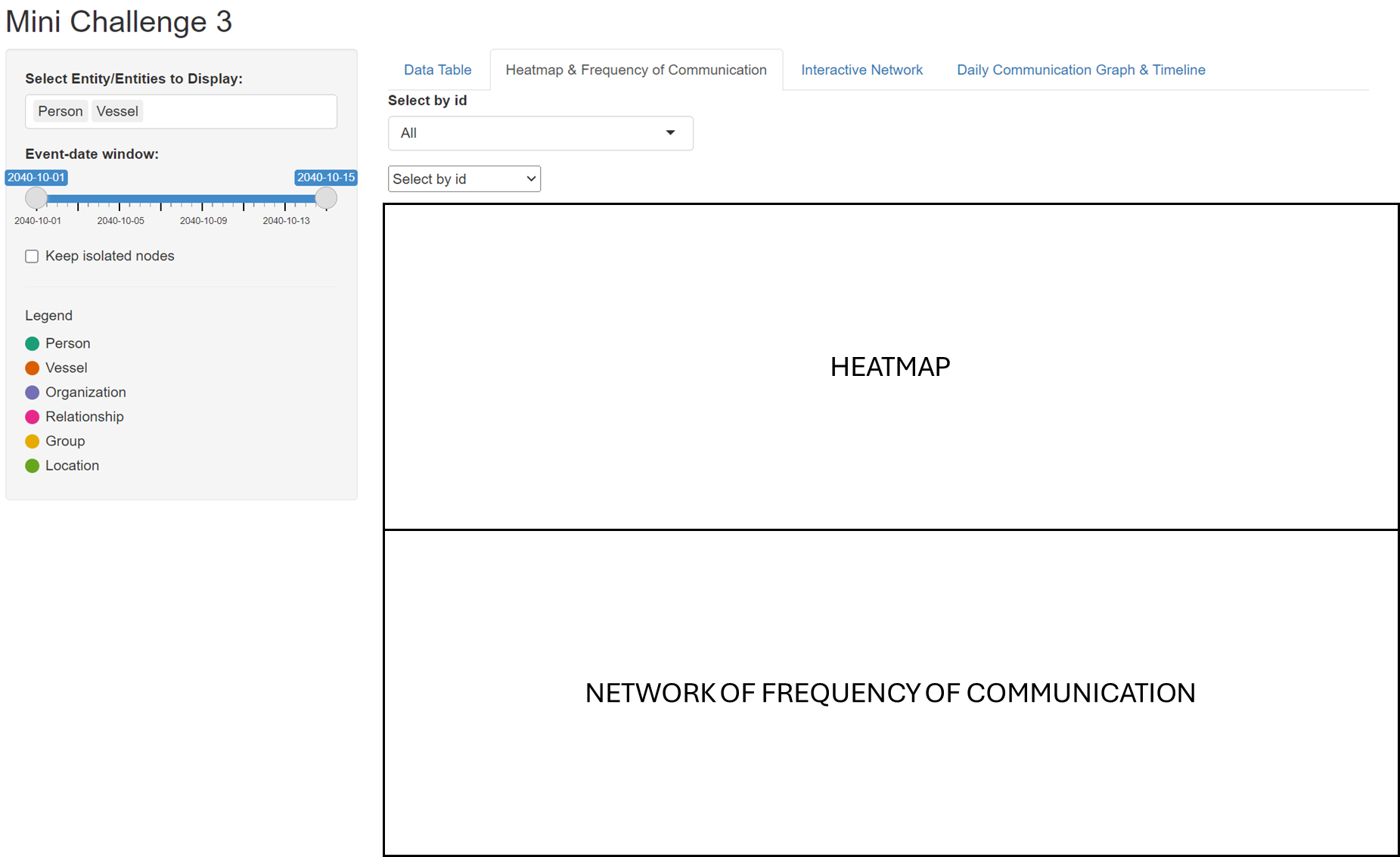
**Overview of Nodes and Edges**

This Data Table gives an overview of the nodes and edges that we are able to use for the visualisations in the other tabs.



### 2nd Tab - Heatmap & Frequency of Communication

The heatmap is able to find out the pattern and frequency of communication between the entities.



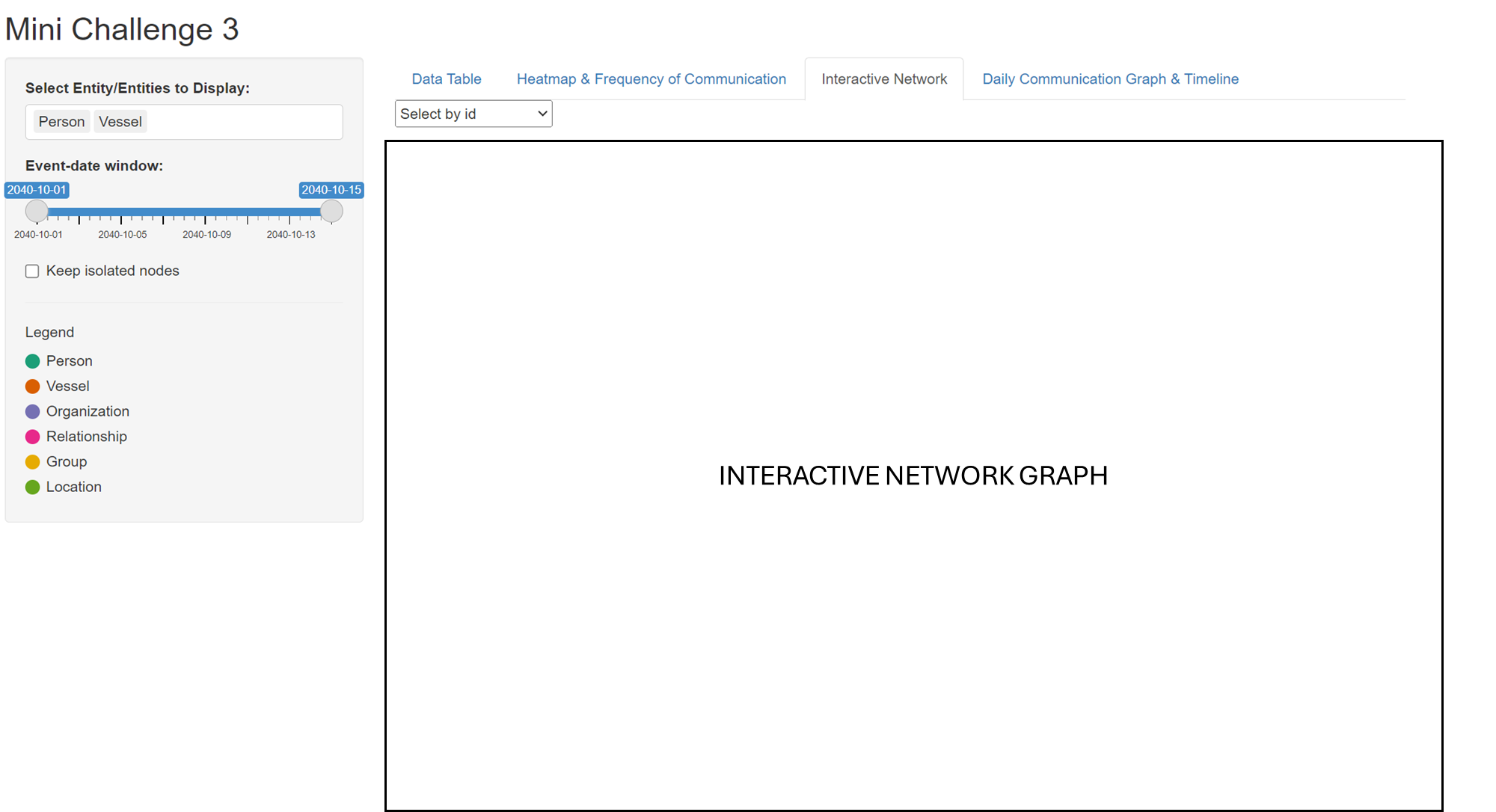
### 3rd Tab - Creating the Interactive Network

The user will be able to select the Entity/Entities and find out the relationship.

The use is also able to toggle the date to narrow down which dates they would like to view.

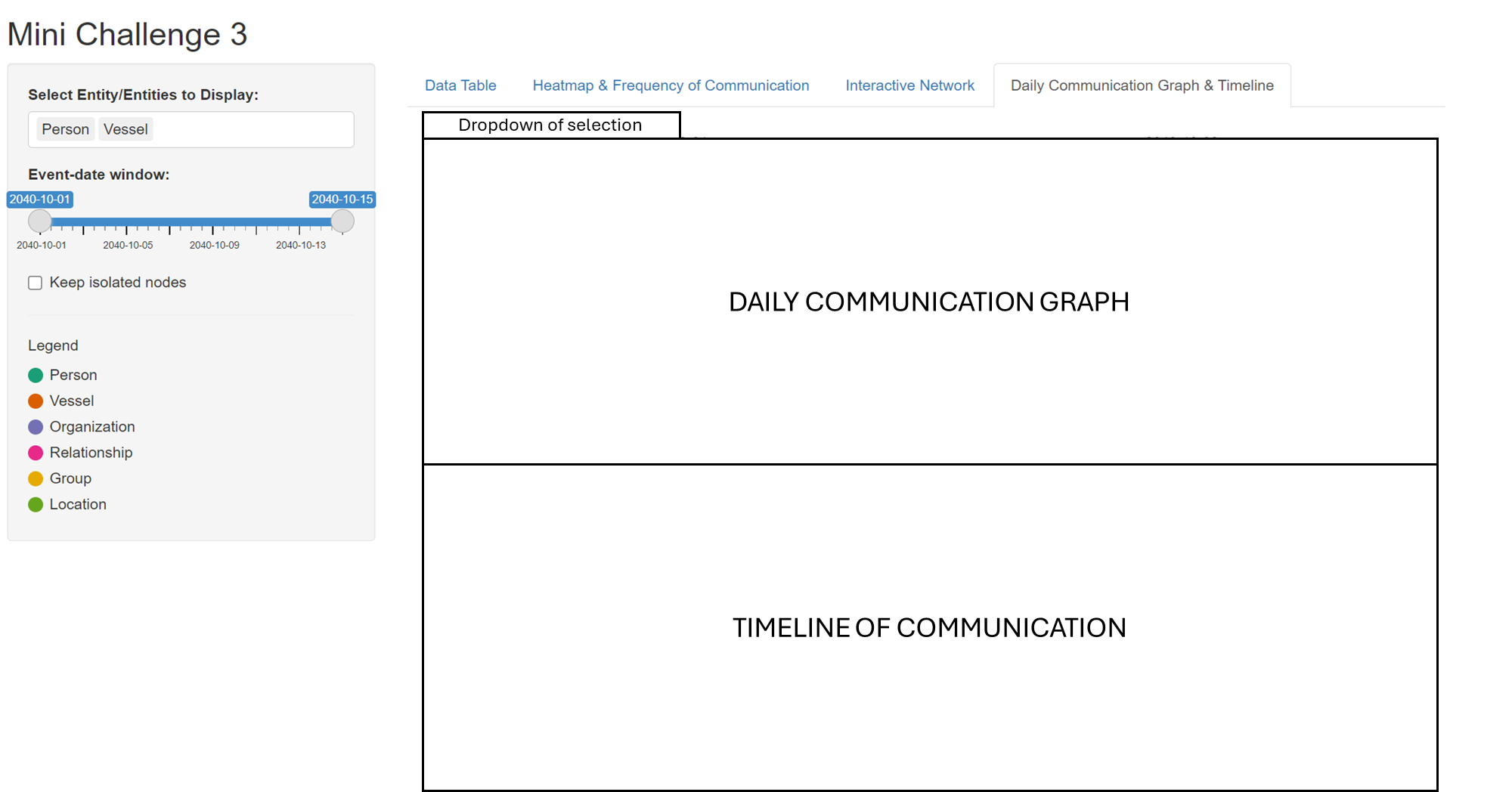
There is also a drop-down if they would like to Narrow down to specific person/relationship.

This network graph will help to figure out the relationship between each person/vessel/organisation/group.



### 4th Tab - Daily Communication Graph and Timeline of conversation

This daily communication graph allows the user to filter the communication for the day and who frequently communicates, narrowing down the groups of people that are associated with each other.



# Conclusion

Overall, the aim of the article is to showcase the initial idea in the development of the Shiny Application that we are planning to build to answer the questions in Mini Challenge 3. This is to allow users with different needs to use this application to filter out relevant necessary information to find out the people who are involved in the illegal activities on Nemo Reef.

The visualization uses colour to differentiate between entity types such as person, vessel, organization, location, or group, which which helps bring clarity to viewers. Additionally, filters are employed to highlight key information, helping users avoid distractions from irrelevant data. We also implemented different data visualization techniques (e.g. heatmap, timeline and network graph) which enables multidimensional exploration, where multiple visual elements complement each other to form a more complete picture of the underlying patterns. Combined with the filters for entities and relationships, users can progressively narrow down potential suspects before zooming in on specific activities and communications. This interactive process empowers users to manipulate and evaluate the network based on the extracted information, facilitating deeper insights and more targeted investigations

# Reference

* [Shiny](https://shiny.posit.co/)
* [ISSS608 AY2024-25 April Term](https://isss608-ay2024-25apr.netlify.app/)