# Clear any existing plots  
plt.close("all")  
  
# Preparing edge data  
edges = []  
for i, path in enumerate(df\_filtered["path"]):  
 nodes = path.split("->")  
 count = df\_filtered["count"].iloc[i]  
 edges += [(nodes[j], nodes[j + 1], count) for j in range(len(nodes) - 1)]  
  
# Creating the graph  
G = nx.DiGraph()  
for source, target, weight in edges:  
 if G.has\_edge(source, target):  
 G[source][target]['weight'] += weight  
 else:  
 G.add\_edge(source, target, weight=weight)  
  
# Positioning nodes for visualization  
pos = nx.spring\_layout(G)  
  
# Drawing the graph  
plt.figure(figsize=(12, 12))  
nx.draw\_networkx(  
 G,  
 pos,  
 with\_labels=True,  
 node\_size=500,  
 node\_color="lightblue",  
 font\_size=10,  
 edge\_color="#888",  
 arrowsize=15,  
 font\_weight="bold",  
)  
  
# Adding edge labels for weights  
edge\_labels = {edge: G.edges[edge]['weight'] for edge in G.edges}  
nx.draw\_networkx\_edge\_labels(G, pos, edge\_labels=edge\_labels, font\_size=8)  
  
# Title and display  
plt.title(f"Directional Network Graph for '{specific\_journey}'", fontsize=16)  
plt.show()