# Use Case Design Specification

## **Recommend a solution to fulfil Request-to-Answer solution with lowest cost**

Step 1: Use the following cypher query to calculate the shortest path for Request-to-Answer solution with lowest cost

#Shortest Path Algorithm (Dijkstra)

**MATCH** (start {name:'Customer1'}), (end {name:'CIR1'})

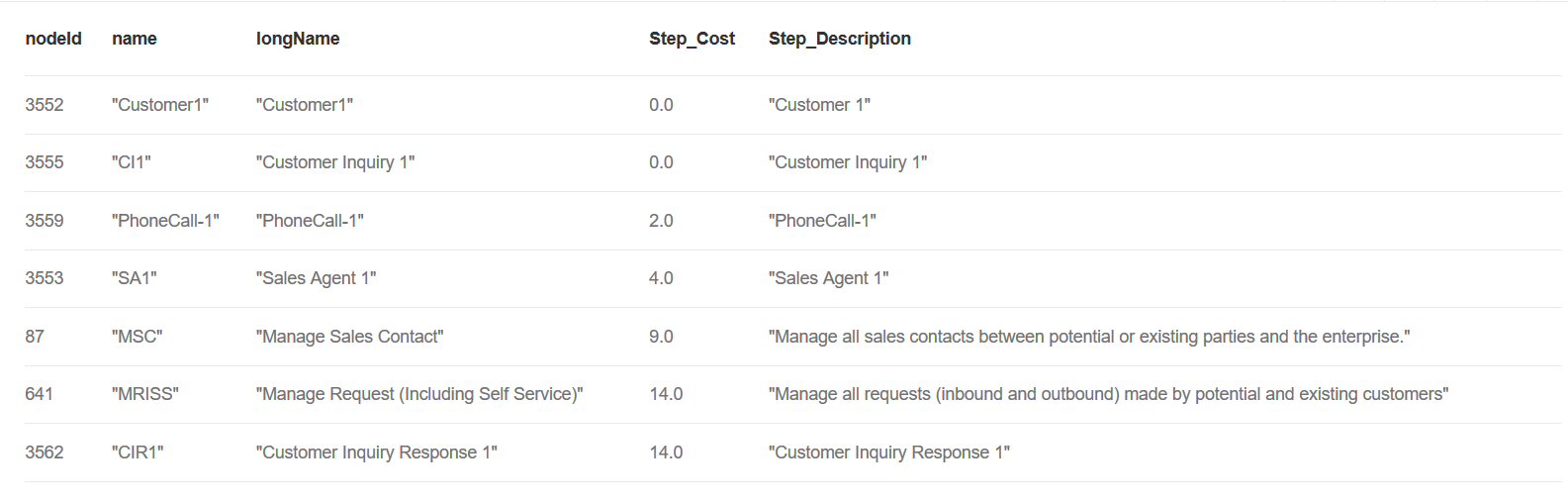
**CALL** algo.shortestPath.stream(start, end, 'relationCost',{direction:'OUTGOING'})

**YIELD** nodeId, cost

**RETURN** nodeId, algo.asNode(nodeId).name as name, algo.asNode(nodeId).longName AS longName, cost as Step\_Cost,algo.asNode(nodeId).shortDescription as Step\_Description

Step 2: Display the following result in the Web UI with one summary statement and a table that illustrate the step cost details

**Solution Summary**: The following steps can fulfil the Request-to-Answer process with the lowest cost 14.0, please refer to below table for the step cost summary

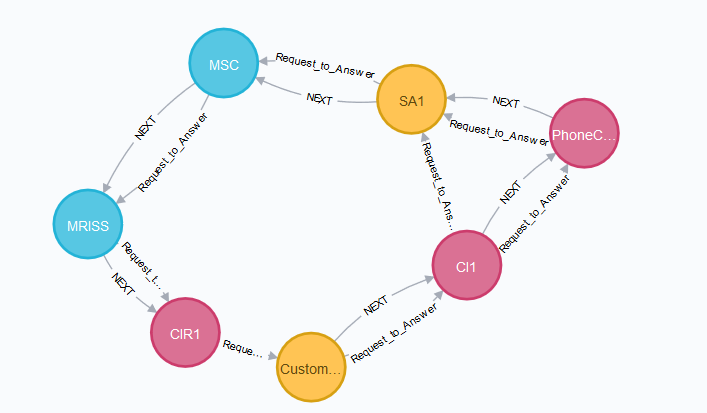


Step 3: use the following cypher query to display the solution in graph view

**CALL** algo.asPath([3552,3555,3559,3553,87,641,3562])

Step4: Display one summary statement and the solution graph view

**Solution Graph View**: The following graph shows the end-to-end graph view about the optimized process flow.



## **Recommend a solution to fulfil Request-to-Answer solution with shortest response time**

Step 1: Use the following cypher query to calculate the shortest path for Request-to-Answer solution with lowest cost

#Shortest Path Algorithm (Dijkstra)

**MATCH** (start {name:'Customer1'}), (end {name:'CIR1'})

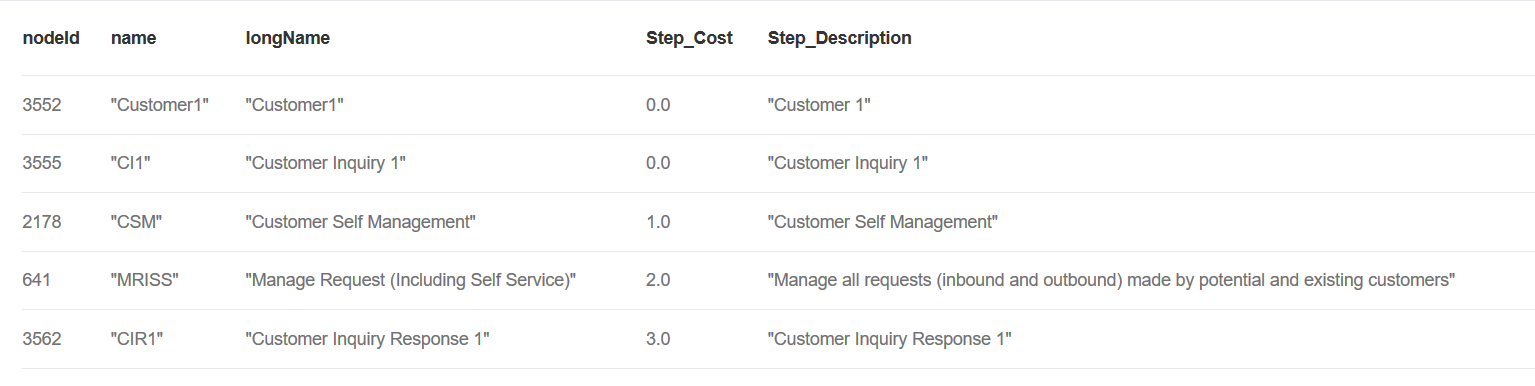
**CALL** algo.shortestPath.stream(start, end, 'relationTime',{direction:'OUTGOING'})

**YIELD** nodeId, cost

**RETURN** nodeId, algo.asNode(nodeId).name as name, algo.asNode(nodeId).longName AS longName, cost as Step\_Cost,algo.asNode(nodeId).shortDescription as Step\_Description

Step 2: Display the following result in the Web UI with one summary statement and a table that illustrate the step cost details

**Solution Summary**: The following steps can fulfil the Request-to-Answer process with the lowest response time 3.0, please refer to below table for the step cost summary



Step 3: use the following cypher query to display the solution in graph view

CALL algo.asPath([3552,3555,2178,641,3562])

Step4: Display one summary statement and the solution graph view

**Solution Graph View**: The following graph shows the end-to-end graph view about the optimized process flow.

