Project Problem #1: OptimalDelivery

OptimalDelivery is a logistics company that wants to move shipments from Producer to Consumer in a distance-optimal manner. They approach your team to help them design a route planning system that enables their fleet of T trucks to pick up from P (predetermined) pickup points, and deliver to D receiving points, so as to incur minimum gas costs, i.e., the fleet has to drive minimum distance collectively.

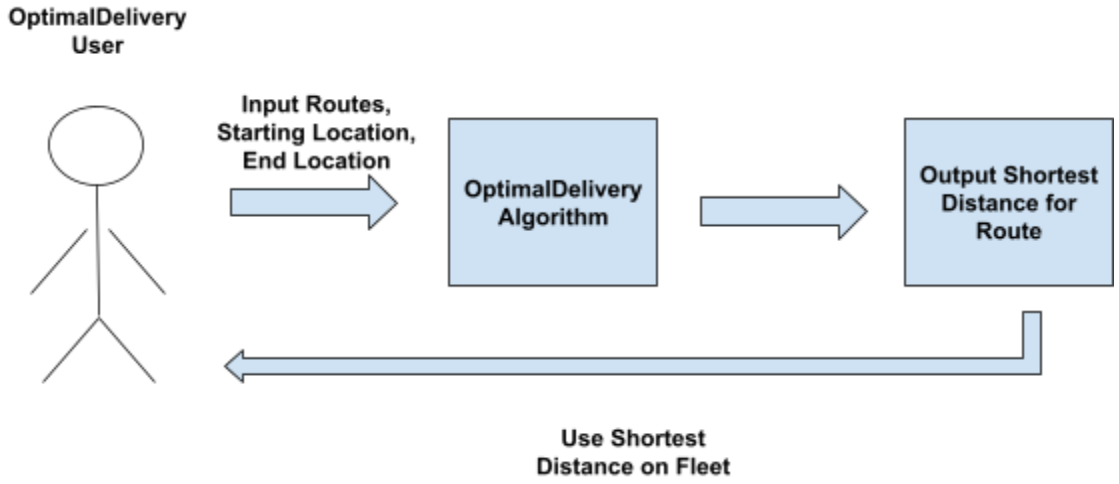
Requirements engineering:

- **User Requirements**
  - Provide the **minimum distance collectively** of the Optimal Delivery fleet that has to drive in order to deliver shipments from **Producer** to **Consumer**
- **System Requirements**
  - Create an Algorithm that finds the shortest distance between a Producer and Consumer
    - Shortest Distance = minimum gas costs
  - Use this algorithm to find each specific shortest distance for the route each truck has to go through
  - Add up the distances of the fleet to ensure it is the minimum distance collectively
- **Functional Requirements**
  - Allow user to input predetermined points of interest:
    - Producer Pickup
    - Consumer Delivery
  - The system shall generate a route from the Producer to the Consumer which will utilize the shortest distance
    - Shortest Distance = minimum gas costs
  - The system will add up the minimum distance collectively to ensure the shortest distance is utilized
- **Non-Functional Requirements**
  - Efficiency Requirements
    - Make Certain the system is always able to output the shortest distance for fleet
      - Incur Minimum gas costs
  - Usability Requirements
    - Make certain system is able to be easily used and understood by OptimalDelivery
      - No Learning Curve, intuitive design
  - Performance Requirements
    - Make certain that the system is able to run and output no matter the input size

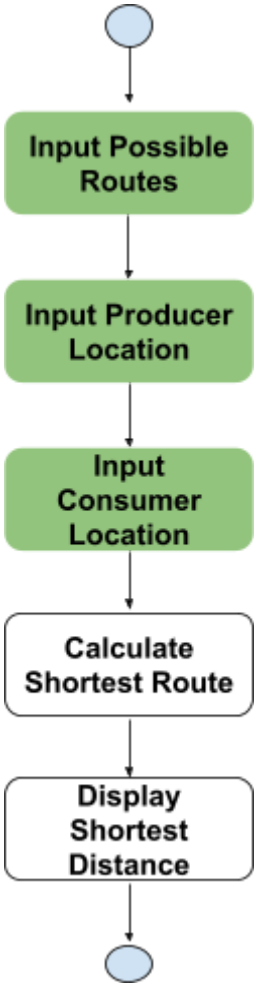
            - With correct input, will always output shortest distance
    - Environmental Requirements
        - Ensure that system is able to output the shortest distance possible
            - Shortest Distance = minimum gas costs
                - Minimum gas consumption = less environmental pollution
    - Safety Requirements
        - Guarantee that routes give by system are safe roads to be driven on
            - Roads can handle traffic of fleet

System Modeling:

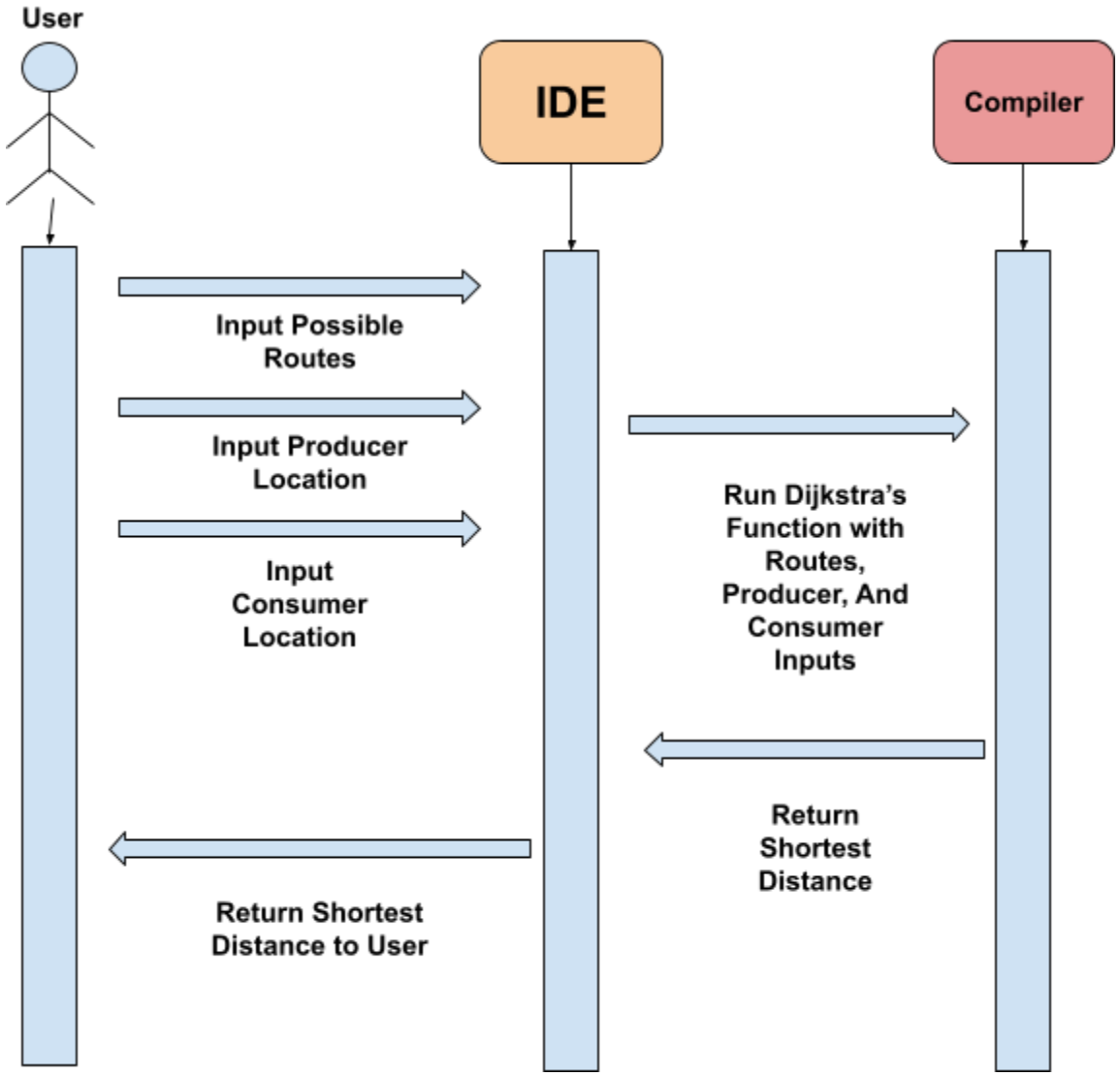
- **Interaction Model:**



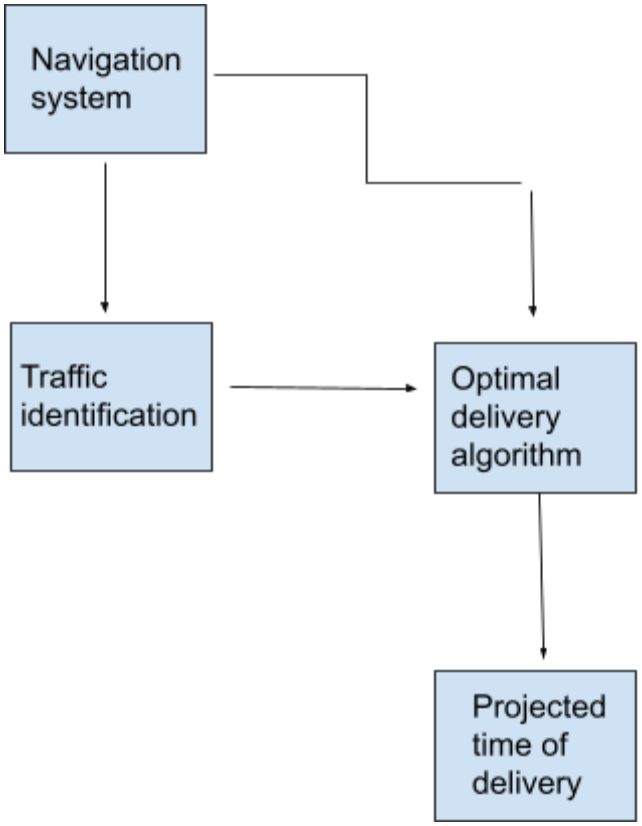
Activity Diagram:



Sequence Diagram:



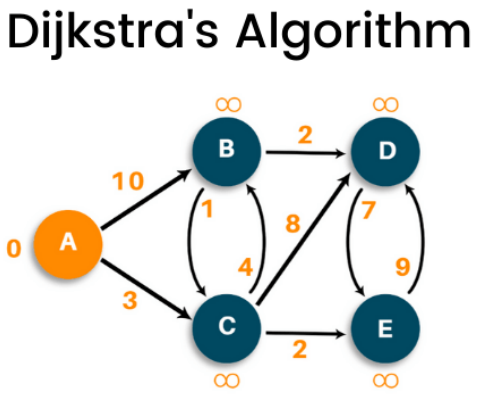
Architectural Design:



add communication(package status)
grouping,

Design and Implementation:

- Make user enter possible routes into graph dictionary
- Input Starting Location, Ending Location (Producer -> Consumer)
- Use dijkstra algorithm to determine the shortest path



- Display Shortest Path to User
- Display Distance to User

Software Testing (across all levels):

- Code not yet constructed as of first sprint

Evaluation:

- Investigated how to implement users requirements
- Through multiple diagrams we determined the ideal and step my step systems the code should follow
- Clear plan of what the desired code should accomplish
- Explored limitations of code and dijkstra algorithm

Project Management:

Description	Impact	Probability	Severity	Mitigation
Team members may not submit code on time or are unresponsive	Deadlines for sprints will have to be pushed. Other team members may need to take up extra work to finish the project on time	Moderate	Serious	Check up periodically before sprint deadline on how everyone is progressing with their assignment
The estimated time for this sprint is underestimated	This would affect the quality of the sprint or it would change the schedule for production of the software	Moderate	Tolerable	Checking up periodically on the progress of each task will help the group adjust the deadlines faster if necessary
Group members do not have the	Team members	Moderate	Tolerable	Tasks should be started early

requisite experience to work on their tasks	would have to learn skills before starting to code their task			so that team members can learn the skills required to complete the task