**Project Problem #1: OptimalDelivery**

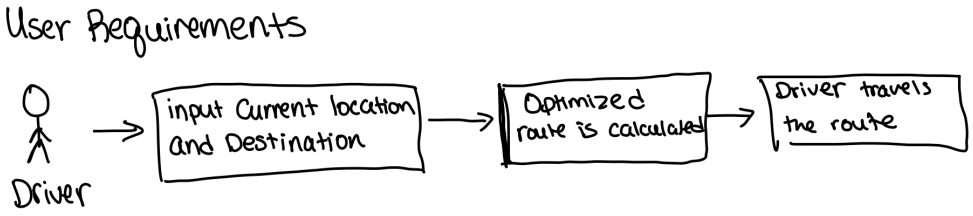
OptimalDelivery is a logistics company that wants to move shipments from Producer to Consumer in a distance-optimal manner. They approach your team to help them design a route planning system that enables their fleet of T trucks to pick up from P (predetermined) pickup points, and deliver to D receiving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

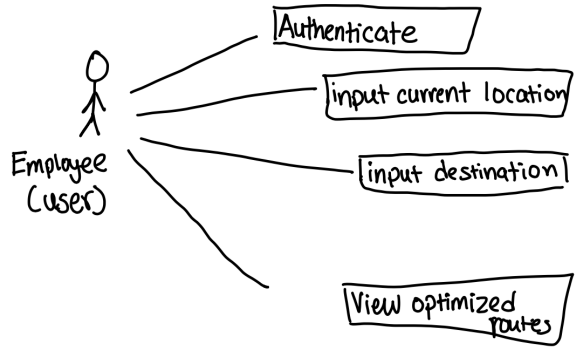
**Requirements Engineering:**

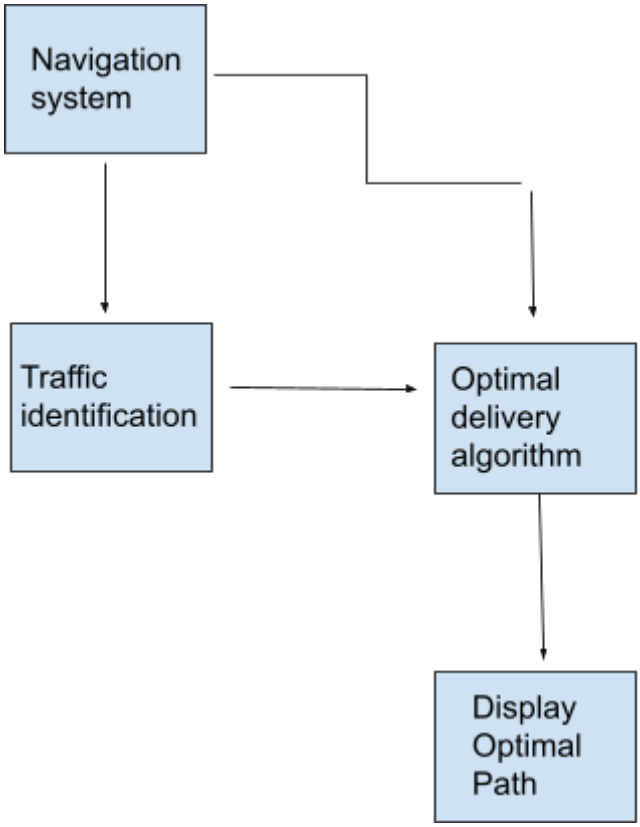
- **User Requirements**
    - The truck drivers (users) will be provided optimal distances from the producer to consumer.
- **System Requirements**
    - New optimal routes will be generated constantly in case of an accident that makes a route unavailable. (using dijkstra's algorithm)
    - In the event of an accident that makes the optimal route unavailable the system will return the next best available route
    - Any employee authorized (i.e fleet manager, driver, etc) can view the trucks routes
- **Functional Requirements**
    - Accept users input, current location and destination
    - The system generates a route from the Producer to the Consumer which will utilize the shortest distance, where shortest distance incurs minimal gas costs.
        - The optimal route is found by comparing the total distances of all the routes between current location and destination.
    - If a route is not available, a message will be displayed
    - This system should not give a false route, a route that doesn't exist or is unavailable
    - Access should only be given to those authorized views of the routes of drivers
- **Non-functional Requirements**
    - Organizational requirement
        - Users need to authenticate themselves by inputting their work id number
    - 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

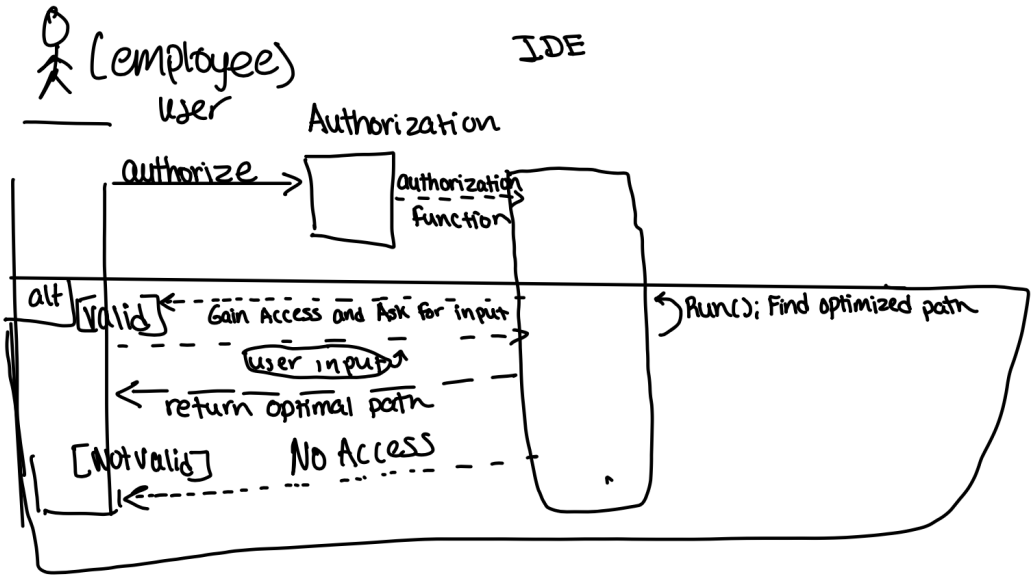
**System Modeling:**



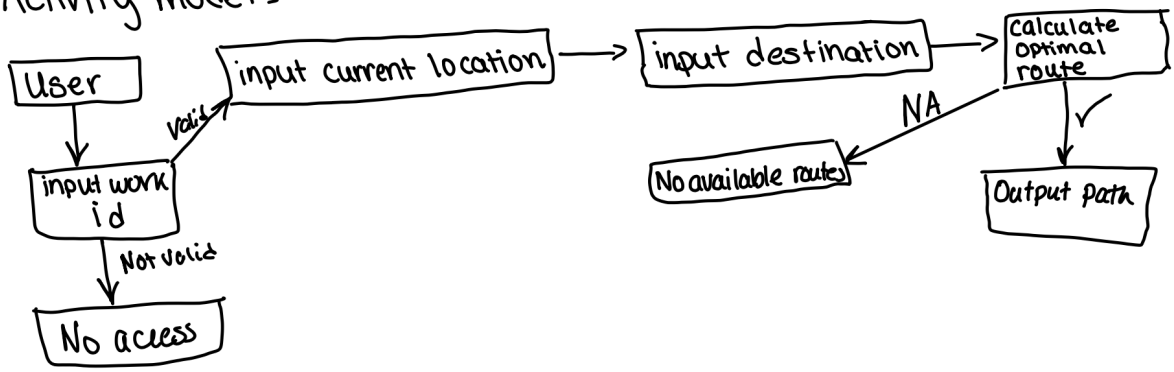



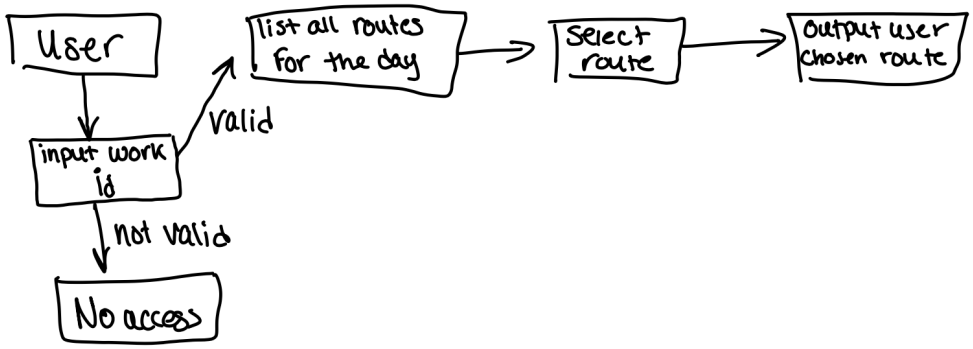



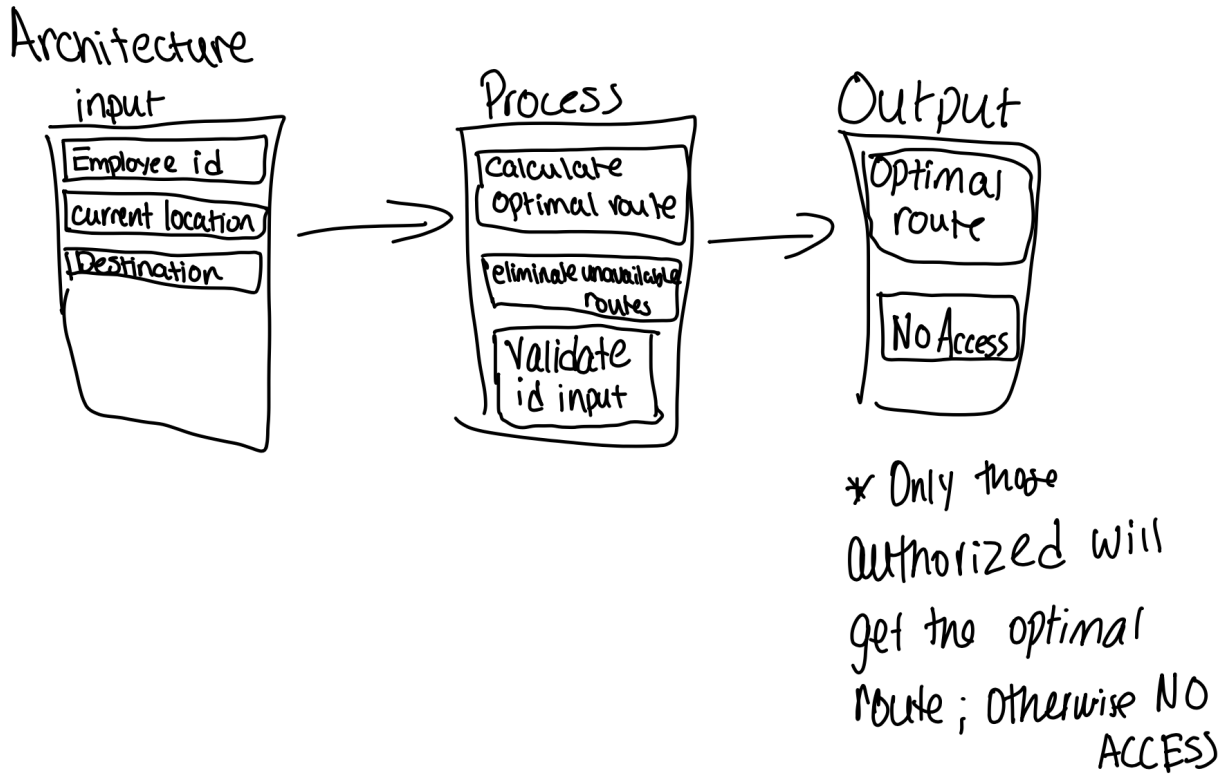
# Sequence Diagram




## Activity models

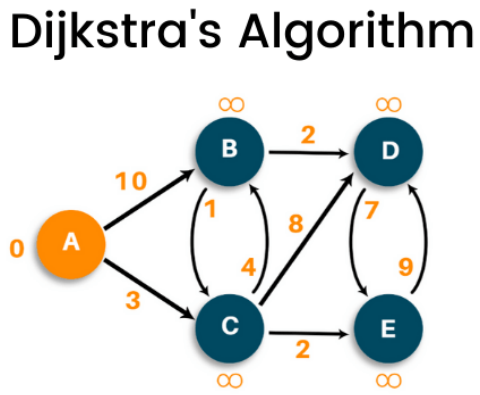





**Design and Implementation:**

- **Algorithm to code: Dijskra's Algorithm**
  - **Make user enter possible routes into graph dictionary**
  - **Input Starting Location, Ending Location (Producer -> Consumer)**
  - **Supposed to find the shortest path which will incur minimal cost**

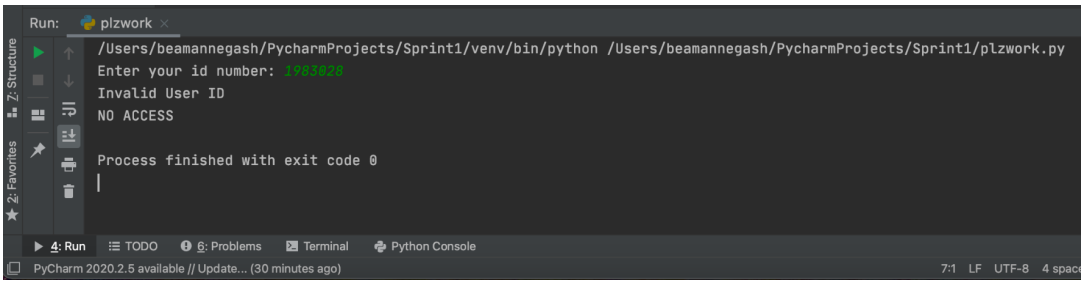



  - 

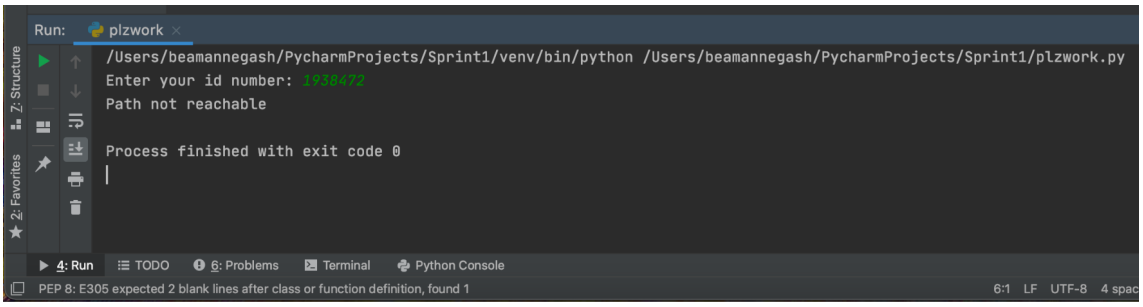
- **Implement authorization function**
  - **Authorizes access to the system, by inputting worker id**
- **Implement random accidents, altering available optimal routes**
  - **Using a random number generator, randomly delete routes**
  - **This can happen at random**

**Software Testing:**

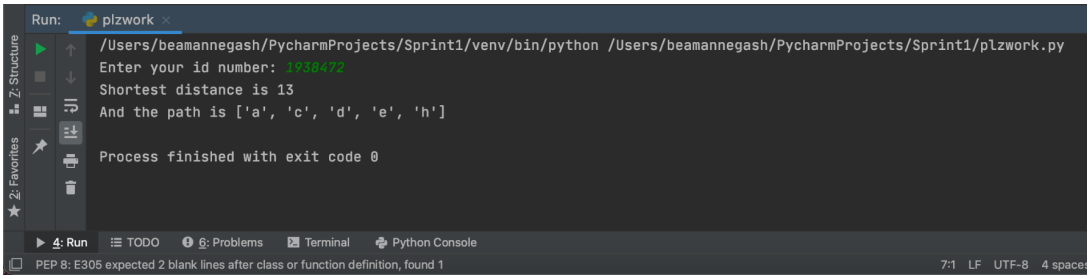
- **Fed the system a list of valid and invalid work id numbers**
  - **The invalid work id numbers were given a "Invalid ID" and "NO ACCESS" message**




  - **The valid work id number were able to use the system**
- **If User inputs a non Valid Starting, End Location, will display "Path Not Reachable"**
  -




- **If User inputs a Valid Starting, End Location, will display the length of the shortest path, as well as path to take**
  -




- **To test the probability of road accident leading to routes being unavailable we changed the probability from 1 in 103 to definite to properly ensure functionality given that random condition**

**Evaluation:**

**We successfully implemented the probability risk factor of car accidents happening and making routes unavailable**

**We successfully implemented an authorization component of the system that only runs for authorized employees via 7 digit work id number**

**Overall we improved the quality of the code and made it more realistic**

**Our next sprint we will implement a GUI and make the requirements more clear and defined**

**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 requisite experience to work on their tasks | Team members would have to learn skills before starting to code their task | Moderate | Tolerable | Tasks should be started early so that team members can learn the skills required to complete the task |