The project follows the guidelines set in the tutorials as well as the review session to take care of the following aspects

* Steering a trajectory
* Detecting objects
* Minimizing jerk and smooth acceleration

The path generation logic includes the lane change as well as speed reduction required to avoid collisions and have a smoother ride.

The first aspect of this is to determine which lane the car is in and to determine if a change to a lane left or right to the existing lane is feasible or not (Eg if you are in lane 2, the right most lane then a right lane change is not possible).

After this we iterate through the list of sensor\_fusion points and take 3 decisions per point.

* The first is to see if an object is present on our current lane and if it is projected to be within 30 meters of the ego car.
* The second is to check that if a left switch was possible, then whether or not the left lane is empty (This is done by checking for an object within 40 meters). The distance is increased to make sure we don’t do any unnecessary lane changes
* The third is to repeat the second step for the right lane.

A flag is set if an object is too close in each of these lanes (*left\_too\_close /right\_too\_close*)

Once we have iterated through the whole sensor fusion list, we check whether the existing lane has an obstacle in front. If so the following actions are taken

* Check if the left change is feasible and the left lane is vacant -> If so, make a left lane change
* If the above step is not possible, check if the right lane change is feasible and the right lane is vacant -> If so, make a right lane change
* If no changes are possible then decrease speed, minimizing jerk by reducing the speed slowly.

If no object is too close in the center lane and if the speed is lower than the required velocity, then speed up with steady and low acceleration.