Approach:-

Struct is made for cabs,riders and paymnet servers which will store data as per requirement.

Threads are created for each rider and payment servers and I have waited until all the riders have paid(if they have to). Once all the riders have paid the program is exited.

I have assigned a variable in struct of cabs which will denote in which state the cab is in and according to that the rider will be assigned the cab.

In Payment servers part, there is a variable which records the number of riders that have paid. If this value plus the riders which did not get a cab equals to total riders then code is exited.

Implementation:-

Struct is made for cabs,riders and payment servers intially.

In Main function:-

Inputs have been taken and intializations have been made.

Then the random generator generates random values for ridetime and waittime and type of cab and a cab is booked by this data.

Each rider is a thread and the function called by this cab books cab for this rider (if it can get a cab).

In Check function:-

Time is noted at the beginning.

Then a loop will go on executing until program exits.

Inside this loop every iteration, time is recorded and if this time minus the start time is greater than the waittime that rider won't get a cab.

In the loop it checks for the type of request.

A cab is allocated after verifying some condition and the data of the struct is updated after a cab is booked.

In Payment function:-

For each rider if it has left the cab a variable in the struct will denote that, so by checking that variable it goes inside the loop after locking that rider and sleeps for 2 sec and again some variables are updated and it comes out of the loop.