**Design Explanation**

**TaxiBookingEntities**

**TaxiFleet** - As described in the problem, taxi fleet consists of three taxi 3, it is singleton object which contains list of taxis.

**Actions:**

                        FindAvailableTaxi

                        ResetTaxis

**Taxi** - Each Taxi is defined by this class which derives from car

**Attributes of Taxi**

                           TaxiId

                           Status

                           Rides of Taxi

                           Taxi Location

**Actions on Taxi**

                             Book Taxi

**Events on Taxi**

                             When Ride starts - Taxi status is changed to Finishing Ride

                             When Ride completes - Taxi is made available

**Ride** - All Rides are associated to individual taxi, Ride subscribe to publisher which publishes time change event

**Attributes of Ride**

                        TimeToReachToSource

                        EstimatedRideTime

                        TimeLeftToReachSource

                        TimeLeftToComplete

                        RideSource

                        RideDestination

**Events on Rides**

                                  When Ride starts - Ride status is changed

                                  When Ride completes - Ride status is marked as completed

**TimeChange Handler** - checks for the time left to reach source and destination

                                   and triggers Ride and Taxi Start and Completed Events.

**SimpleTimeTracker** – Tracks the current time from epoch time and Publish message indicating time change.

**WorldMap** – And singleton instance which returns map for city (This is more to cater any future request to have map for specific city rather than complete world map)

**Location -** Used for source destination and Taxi location

**TaxiBookingSystemCore -** Facilitates the execution of taxi booking service

**TaxiBookingSystemUnitTests** - Covers the test scenarios for booking taxi and ride status

**TaxiBookingSystemCommons** – Consist of Extensions and common application logic used across different projects.

**TaxiBookSystem –** the api project which uses kestrel as web server and have TaxiBookingController which consist of restful endpoints, you can use swagger to understand details of endpoints and Model used in project.