In this project you are asked to create a smart contract that handles a common asset and distribution of income generated from this asset in certain time intervals. The common asset in this scenario is a taxi. Imagine a group of people in the same neighborhood who would like to invest into an asset. They cannot invest individually because each person has a very small amount of money, thus they can combine their holdings together to invest into a bigger and more profitable investment.
They decided to combine their money and buy a car which will be used as a taxi and the profit will be shared among participants every month. However, one problem is that they have no trust in each other. To make this investment work, you are asked to write a smart contract that will handle the transactions. The contract will run on Ethereum network.
The contract should at least include below. If you need to extend the state variables and functions, you are free to do so as long as they are necessary. Do not forget that storage is expensive on Ethereum. For function parameters try to write functions as few parameters as possible, none preferred if possible.
Implementation : Use Solidity language on Remix.

State Variables:

Participants: maximum of 100, each participant identified with an address and has a balance
Manager: a manager that is decided offline who creates the contract initially, can be changed later
Taxi Driver: 1 taxi driver
Car Dealer: An identity to buy/sell car, also handles maintenance and tax
Contract balance: Current total money in the contract that has not been distributed
Fixed expenses: Every 6 months car needs to go to Car Dealer for maintenance and taxes needs to be paid, total amount for maintenance and tax is fixed and 10 Ether for every 6 months.
Participation fee: An amount that participants needs to pay for entering the taxi business, it is fixed and 100 Ether.
Owned Car: identified with a 32 digit number, CarID
Proposed Car: Car proposal proposed by the CarDealer, Holds {CarID, price, and offer valid time} information.
Proposed Purchase: Car purchase proposal proposed by the CarDealer, Holds {CarID (the owned car id), price, offer valid time, and approval state} information.
Time handles

Functions:

Constructor:
Called by owner of the contract and sets the manager and other initial values for state variables
Join function:
Public, Called by participants, Participants needs to pay the participation fee set in the contract to be a member in the taxi investment
SetCarDealer:
Only Manager can call this function, Sets the CarDealer’s address
CarPropose:
Only CarDealer can call this, sets Proposed Car values, such as CarID, price, and offer valid time
PurchaseCar:
Only Manager can call this function, sends the CarDealer the price of the proposed car if the offer valid time is not passed yet.
PurchasePropose:

Only CarDealer can call this, sets Proposed Purchase values, such as CarID, price, offer valid time and approval state (to 0)

ApproveSellProposal:

Participants can call this function, approves the Proposed Purchase with incrementing the approval state. Each participant can increment once.

Sellcar:

Only CarDealer can call this function, sends the proposed car price to contract if the offer valid time is not passed yet and approval state is approved by more than half of the participants.

SetDriver:

Only Manager can call this function, sets the Driver info

GetCharge:

Public, customers who use the taxi pays their ticket through this function. Charge is sent to contract. Takes no parameter.

PaySalary:

Only Manager can call this function, releases the salary of the Driver to his/her account monthly. Make sure Manager is not calling this funciton more than once in a month.

GetSalary:

Only Driver can call this function, if there is any money in Driver's account, it will be send to his/her address

CarExpenses:

Only Manager can call this function, sends the CarDealer the price of the expenses every 6 month. Make sure Manager is not calling this function more than once in the last 6 months.

PayDividend:

Only Manager can call this function, calculates the total profit after expenses and Driver salaries, calculates the profit per participant and releases this amount to participants in every 6 month. Make sure Manager is not calling this function more than once in the last 6 months.

GetDividend:

Only Participants can call this function, if there is any money in participants' account, it will be send to his/her address

Fallback Function: