

BBM 443 Project-Fall 2020
Instructor: Dr. Adnan Özsoy
Email: adnan.ozsoy@hacettepe.edu.tr
Shared Taxi Business on Blockchain
Due: 03 Jan 2021 – NO EXTENSION



Courtesy of <https://coindelite.com/news/blockchain-in-taxi-new-technology-for-common-people/>

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. For function parameters try to write functions as few parameters as possible, none preferred if possible.

For the functions that send and receive money, like join, purchasecar, sellcar, getchargei getsalary, carexpenses, getdividend should send/transfer the amount to the address that it supposed to. you can keep a mapping for the balance for individuals, and update the balances internally but when above functions called, you need to actually send/receive money (ether/wei).

State Variables:

Participants: maximum of 9, each participant identified with an address and has a balance

Manager: a manager that is decided offline who creates the contract initially.

Taxi Driver: 1 taxi driver and salary

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, offer valid time and approval state } information.

Proposed Repurchase: Car repurchase 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

CarProposeToBusiness:

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

ApprovePurchaseCar:

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

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 and approval state is approved by more than half of the participants.

RepurchaseCarPropose:

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 Sell with incrementing the approval state. Each participant can increment once.

Repurchasecar:

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.

ProposeDriver:

Only Manager can call this function, sets driver address, and salary.

ApproveDriver:

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

SetDriver:

Only Manager can call this function, sets the Driver info if approval state is approved by more than half of the participants. Assume there is only 1 driver.

FireDriver:

Only Manager can call this function, gives the full month of salary to current driver's account.

PayTaxiCharge:

Public, customers who use the taxi pays their ticket through this function. Charge is sent to contract. Takes no parameter. See slides 6 page 11.

ReleaseSalary:

Only Manager can call this function, releases the salary of the Driver to his/her account monthly. Make sure Manager is not calling this function 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

PayCarExpenses:

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:

A note on car buying/selling process:

First dealer calls "CarProposeToBusiness". Participants vote to approve the proposed car through "ApprovePurchaseCar". If the approval state is approved by more than half of the participants then manager calls "PurchaseCar".

To sell the car, manager and dealer talks out side of the system in person. If the dealer agrees to buy the car he/she calls the "RepurchaseCarPropose" function with intended price for the car. Participants vote to sell through "ApproveSellProposal". If majority approves, car dealer calls "Repurchasecar".

Implementation: Use Solidity language on Remix to implement above functionality.

Discussion question:

Above model requires a manager to work. Is it possible to propose a model without the need of a manager?

If yes, explain which functions need to be changed and how.

If no, explain thoroughly.

Write a short answer (1-2 paragraph) for the discussion question and include as a PDF document to the submission.

Grading:

20 functions – 4,5 points each

Discussion – 10 points

Submission:

Include a README file that has your name, ID and if you need to mention any necessary info.

Send taxi.sol file, discussion.pdf, and README through submit.cs system.