

BBM 443 Project-Fall 2021
Instructor: Dr. Adnan Özsoy
Email: adnan.ozsoy@hacettepe.edu.tr
Shared Taxi Business on Blockchain
Due: Will be announced thru Piazza



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).

Keep a contract balance.

State Variables:

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

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 initial values for state variables (like CarDealer)

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

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. When the majority of participants approve, calls PurchaseCar function

PurchaseCar:

Sends the CarDealer the price of the proposed car if the offer valid time is not passed yet

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. When the majority of participants approve, calls Repurchasecar function

Repurchasecar:

Sends the proposed car price to contract if the offer valid time is not passed yet

ProposeDriver:

Caller proposes himself/herself as driver and sets his/her address, and expected salary. Only one proposed driver can be set,

ApproveDriver:

Participants can call this function, approves the Proposed Driver with incrementing the approval state. Each participant can increment once. When the majority of participants approve, calls SetDriver function

SetDriver:

Sets the Driver info if approval state is approved by more than half of the participants. Clears proposed driver info

ProposeFireDriver:

Participants can call this function, approves firing driver with incrementing the approval state. Each participant can increment once. When the majority of participants approve, calls Fire Driver function

FireDriver:

Sends any amount of money of the current driver's account to Driver's address. Clears the driver info

LeaveJob:

Only Driver can call this function, calls FireDriver function.

GetCharge:

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.

GetSalary:

Only Driver can call this function, releases the salary of the Driver to his/her account monthly. Make sure it is not being called more than once in a month. If there is any money in Driver's account, it will be sent to his/her address.

CarExpenses:

Only one of the participants can initiate this function, sends the CarDealer the price of the expenses every 6 month. Make sure it is not being called more than once in the last 6 months.

PayDividend:

Only one of the participants can initiate 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 it is not being called more than once in the last 6 months. This does NOT send money directly, but only updates the balance on the contract and participants.

GetDividend:

Only Participants can call this function, if there is any money in the calling 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 function calls "PurchaseCar".

To sell the car, dealer talks to people outside 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, "Repurchasecar" is called.

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

Grading:

20 functions – 5 points each

Submission:

Zip everything and send through SUBMIT system. Include a README file that has your name, ID and if you need to mention any necessary info.