Free University of Bolzano Bozen – Faculty of Economics and Management

Big Data and Blockchain exam

# Rules

* + No communication with other people or among students or with artificial intelligences is allowed. Phones and every other means of communication must be turned off. Opening any communication program on the computer is not allowed and is considered cheating.
  + You are responsible for the correct copy of your files.

## Exercise Smart Contracts

You have 70 minutes starting from now.

## Idea

Your task is to build a contract to handle the “shared” organization of group trips for a tour operator. The tour operator starts a new trip offer with the indication of minimum and maximum amount of passengers and the per-person price. Anyone can participate the trip by making some down-payment and indicating the amount of people they want to add to the trip. At departure time, the operator verifies if the conditions for departure are met, gets payed, and starts the trip. At any moment, anyone has the opportunity to chip in all the missing money due to an insufficient number of participants to allow the start of the trip.

## Task

Write Solidity contract **grouptrip**.

## Suggested data structures

    address payable private operator;

    uint public startTime;

    uint public fee;

    uint public minPax;

    uint public maxPax;

    uint public people;

## Step 1

Build a **constructor** which sets operator, minPax, maxPax and fee. It also sets people and startTime to 0.

## Step 2

Add a function **sign\_up**. The user must indicate the number of passengers and pay an amount of wei equal or larger than fee by number of passengers. If he pays more, too bad for him. Be sure to verify that the trip has enough space. Adjust variable people.

**FLIP PAGE **

## Step 3

Modify the previous function in such a way that at the time of first sign-up it also stores the startTime to be within the next 40 seconds. If instead it is not the first sign-up (check it with an IF on startTime) it does not modify it.

## Step 4

Add a function **start\_trip**, which is usable only by the organizer when the current time is at least startTime. If the criteria for the operator are fulfilled in terms of funds (fee \* minPax), the function pays the operator and returns **true**. Otherwise, it sets a new start time of now + 40 seconds and it returns **false**.

## Step 5

Deploy the contract on Sepolia’s blockchain and test all the previous functions using Account 1 for the operator and Account 2 an applicant. *If you did not manage to do it fully, after having copied everything into a text file, remove the problematic functions and deploy and test what you have.*

## Step 6

Add a view **missing\_balance**. It returns a the amount of the travel fee missing to cover the minimum amount of passengers (fee \* minPax).

## Step 7

Add function **chip\_in** which works only if there is at least one booked person. Through this function anybody can add funds to the contract to cover the missing amount of wei to reach the minimum amount of minPax by fee (you can get this value using missing\_balance() function above).

## Save and return:

* a text file **grouptrip.txt** containing the Solidity source code that you have built
* your Sepolia’s contract’s address