COMP6451 Summer 2018-19
Assignment 2 - Ethereum Programming
Total Marks: 20
Due: 23:00 Sun Feb 10

Techno-Groovy Band, the latest sensation in techno-pop music, is holding a large outdoor concert, and 200,000 are expected to attend, though it is expected that as many 500,000 will try to buy a ticket. To demonstrate to their fans just how futuristic they really are (in spite of their name suggesting they date from the 1960's!), they have decided to accept payment in Ethereum and issue tickets as crypto-tokens on the Ethereum blockchain, called GroovyTix.

Techno-Groovy Band would like to protect their fans against a number of situations that lead to bad experiences for concert goers: inability to attend due to illness or unexpected obligations, weather washouts and ticket scalpers. With this in mind, they have set the following requirements for the system.

1. The concert date is Mar 1, 2019.
2. A maximum of 200,000 tickets are to be sold.
3. Tickets cost 0.5 Ether.
4. Ticket sales open at midnight on Feb 15, 2019 and close at midnight on Feb 28 2019.
5. To prevent scalpers acquiring tickets, the ticket system should issue at most one ticket to each fan.

6. There should be a process whereby a fan uses their smart-phone to prove that they hold a ticket, for entry to the concert.

7. To prevent a ticket being used for entry to the concert area by multiple people at the same time, the system should maintain a record of whether a ticket holder is inside or outside of the concert area. This may require a process enabling "checking out" of the concert area once a fan has entered.

8. A fan should be able to transfer their ticket to a friend, in exchange for 0.5 Eth, prior to Feb 20. As an anti-scalping measure, this payment should be mediated by the smart contract, to make sure the amount is not larger than permitted.

9. As an alternative to transferring a ticket to a friend, a user should be able, before Feb 27, to request a refund.

10. Refunds will be granted provided that there is another fan who missed out on a ticket and has requested and pre-paid for a "rush" ticket.

11. "Rush tickets" will be allocated on a first-come, first-served basis. Requesters of a "rush" ticket who do not receive an actual ticket by the time of the concert will get a refund of their pre-payment.

12. There should a function callable by Techno-Groovy Band, that refunds all ticket holders in case of a concert washout.

13. After the concert, assuming no washout, Techno-Groovy Band should be able to withdraw the concert proceeds to their own account.

14. To provide fans maximum confidence that the ticketing process will abide by the above rules, as much of the system as possible should be be implemented using an Ethereum smart contract that enforces the rules.

You have been retained by Techno-Groovy Band to develop an Ethereum based system to implement GroovyTix.

Part 1 (15 marks): (This part may be done entirely using Remix) You should submit the following:

- A copy of Solidity smart contract code to be used in a system that addresses the above requirements.

- A report to the client, describing your *design and analysis* of the overall system. This report should explain how each of the above requirements has been addressed, and what you have done to test your implementation. You should try to comply with the requirements as much as possible. However, in case there are any issues with any of the requirements, your report should explain these issues to the client, and describe any changes you have made.

- A copy of the test cases you have used to test your smart contract code.

You should endeavour to write the smart contract so that it is easily reusable for other concerts with similar requirements in the future. You do *not* need to develop user interface components of the system, such as a web-page for buying a ticket, the mobile apps used by fans to attend the concert, or the app used by concert staff to manage admission to the concert. However, your report should explain how these will interact with your smart contract in order to deliver the required functionality.

You do not need to have deployed the contract onto the live Ethereum blockchain. However, you should include in your report an analysis of the security and expected performance of the system. In particular, evaluate how well your system can be expected to operate on the public Ethereum blockchain once it is deployed, particularly with respect to throughput in the peak periods of ticket buying once tickets are first released, as well as during the course of the concert.

Given your design, are there any ways that malicious users might try to circumvent any of the design requirements? If so, explain what they are, and any operational measures that might be taken to mitigate them.

Part 2 (5 marks): The above requirements leave it to Techno-Groovy Band to make the decision about whether there has been a washout. Some fans might worry that Techno-Groovy Band will take the money and run even if there has been a washout. The following alternate version of this part of the requirements would address this concern:

1. The smart contract should have a function that takes as input a signed weather report message. (Note that signing cannot be done by the

smart contract, because that would reveal the private signature key on-chain, where it can be seen by anyone.) The signature on this message should verify using the public key belonging to a trusted weather reporting service. If the message states “Rainfall at the Techno-Groovy Band concert was: r mm”, where $r > 10$, then all ticket holders should be refunded 0.5 Eth.

2. Any Ethereum user should be able to submit the weather certificate as evidence.

3. If the weather certificate has not been submitted by Mar 7, then the Techno-Groovy band should be able to withdraw the balance of all funds remaining in the smart contract. To assure fans that they will not be ripped off, a withdrawal by the Techno-Groovy band prior to this date should not be possible.

For this part of the assignment, implement the above functionality, providing verification code in the smart contract as well as off-chain code for constructing the signed message at the weather service. This part requires that you investigate how to sign messages off-chain and have a smart contract verify it. Consider the Solidity function `ecrecover`, and how to construct signed messages in frameworks such as Web3.py or Web3.js. Include in your report a description of what you have done for this part.