Part 1

Create and deploy a token that is completely owned by yourself - you are the issuer. Assume there are 5 wallets in this network. All wallets on the network are initially whitelisted. Whitelisted wallets can transfer tokens to other whitelisted wallets. But blacklisted wallets do not have the privilege to send or receive tokens. Create functionality to blacklist wallets. If a wallet is blacklisted, then all peers that have directly interacted with this wallet also get blacklisted. If a blacklisted wallet is re-whitelisted, then peers that were blacklisted because of it, also get whitelisted.

Objectives

- Create a UML diagram (or equivalent) to showcase how you would architect such a system
- Create smart contracts to achieve this goal - you may use remix to deploy them on a testnet of your choice
- Create a front-end app for users to purchase tokens at the fixed rate
- Create a front-end app for the token owner to whitelist or blacklist addresses
- List out any edge cases that you think may be pertinent and how you would solve them

Part 2

What happens when you increase the number of wallets on the network to 1000? Does a smart contract-only approach still work? If not, what alternative solution do you propose?

Objectives

- Create a UML diagram (or equivalent) to showcase how you would architect such a system
- List out any edge cases that you think may be pertinent and how you would solve them

Timeline
~3 days