## Exercise Prompt

**Write the Solidity architecture (function names and their inputs and outputs) for the smart contracts for your final project. Upload it to the Github repo you made in Chapter 1.**

## Exercise Concept

Since the last exercise, you've learned:

* Background of the Ethereum network
* How accounts, transactions and state is stored on the Ethereum network
* How to run an Ethereum client
* How Smart Contracts fit into our blockchain mental model
* How to use Truffle
* Solidity fundamentals
* Smart contract design patterns
* Security pitfalls and attack vectors
* Options for checking and protecting your application

For this exercise, we'd like you to take your understanding of smart contracts and apply it to your final project.

Specifically, we'd like you to think about the **single workflow for the future user of your project** you wrote in the first exercise. For that workflow, write out the basic smart contract functions that will be required. You don't have to write the Solidity logic yet!

## Exercise Parts

**1. As a first step, just write the function name, what inputs the function requires and what it might return. Start a new file in your Github repo and start to sketch the functions.**

Here's what that might look like for our voting example from the first exercise:

*1. Users will have to register themselves somehow on the contract*

function registerVoter(address \_voter) {

// registers voter

};

*2. They have to identify which campaign their voting on*

function registerVote(uint campaignID) {

// registers the vote of the voter

};

*3. They'll have to submit a vote for that campaign but 4. they can't vote twice for a single campaign.*

modifier onlyVoteOnce() {

// checks the vote hasn't voted before

\_

};

Again, you don't have to write the internal logic quite yet (although you can). **The point is to have to start to think about the user's behaviors in terms of discrete actions and what those actions require.** You can see in the example above how we already start to have a sense of some global variables the contract will require to execute these functions. That's the point of this exercise, **to start understanding the general contours of how a user will interact with your smart contract code.**

**2. Once you have four to five functions sketched out, commit and push the change to your existing final project Github repo.** Once you've pushed to the repo, share your link with your study group or sessions or in the Discord!

[**https://github.com/YOUR\_GITHUB\_USERNAME\_HERE/education-dao-final-project**](https://github.com/YOUR_GITHUB_USERNAME_HERE/blockchain-developer-bootcamp-final-project)

Happy sketching!