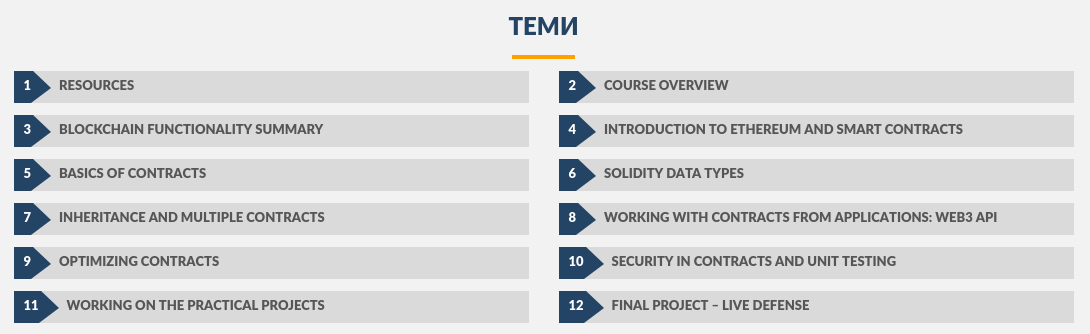
# Smart Contract Development with Solidity - Project Requirements

Your project is what you will be judged and scored based on. You are free to develop it at any time using any learning resources, but do not copy each other’s work. At the end of the course, you will be asked to present and defend your project. You should explain what you built and what thoughts made you take certain choices about the architecture and similar. You can also explain what work you did in the chronological order.

Projects will be scored based on project requirements. All main (‘mandatory’) requirements sum up to 100% score. There are optional tasks that will give you extra if you choose to make a greater impression.

**Each requirement has a lecture number. This means that you should be able to do this task after that lecture**. The numbers are according to the following list:



There are **two** proposed project **topics**, from which you can **choose** **one**:

* **Decentralized Domain Name System (DDNS)**
* **Marketplace contract**

## DDNS

### Mandatory Tasks

* **(20%) (lecture 6)** Public method to register a domain, giving the domain name and an ip address it should point to. A registered domain cannot be bought and is owned by the caller of the method. **The domain registration should cost 1 ETH and the domain should be registered for 1 year**. After 1 year, anyone is allowed to buy the domain again. The domain registration can be extended by 1 year if the domain owner calls the register method and pays 1 ETH. The domain can be any string with length **more than 5 symbols**.
* **(10%) (lecture 6)** Public method to edit a domain. Editing a domain is changing the ip address it points to. The operation is free. Only the owner of the domain can edit the domain.
* **(10%) (lecture 6)** Public method to transfer the domain ownership to another user. The operation is free. Only the domain owner can transfer his domains.
* **(10%) (lecture 6)** Public method to receive an IP based on a given domain.
* **(10%) (lecture 6)** Public method that returns a list of all receipts by a certain account. A receipt is a domain purchase/extension and contains the price, timestamp of purchase and expiration date of the domain.

## Marketplace Contract

### Mandatory Tasks

* **(20%) (lecture 6)** Public method to buy a store item by specifying its ID (a string) and quantity. The method should execute successfully if the marketplace has enough of the item in stock and the sent funds are sufficient. Overpay is considered a tip.
* **(10%) (lecture 6)** Public method to update the stock of an item by taking its ID and the new availability (items in stock). **This method should only be called from the contract owner**!
* **(10%) (lecture 6)** Public method to add a new product to the Marketplace by specifying its name (string), price and initial quantity. Only called by the contract owner.
* **(10%) (lecture 6)** Public method to get the price, name and stock about a product by its ID.
* **(10%) (lecture 6)** Public method to that returns an array of all product IDs.

## Common Tasks

#### Mandatory Tasks (for both topics):

* **(40%) (lecture 10)** **Unit tests for all the methods in your contract** (including all aforementioned). The tests should handle **all constraints around the contract**. Example with DDNS: A test can be one that tries to register an already registered domain. The test is passed if the operation fails (expected behavior).

#### Optional Tasks (for both topics):

* **(5%) (lecture 5)** Use **contract events** to signify that an activity has taken place in your contract. Events can be for domain registration / transfer (DDNS) or item purchase / stock update (Marketplace), for example.
* **(20%) (lecture 8)** Create a basic website with MetaMask that connects to a contract (published in a test net or local blockchain). The application should allow **at least one** operation with the contract (Domain registration or Store purchase are examples).
* **(5%) (lecture 6)** Dynamic pricing. For DDNS, the base price can increase if a short domain name is bought. For Marketplace, price can increase as the stock of an item lowers.
* **(5%) (lecture 6)** Public method to withdraw the funds from the contract. **This should be called only from the contract owner** (the address which initially created the contract).

## Submission

Submission deadline: **29.03.2018 23:59**

Projects should be submitted as archive files on the course’ page under the “Final Project - Live Defense” heading. A button for uploading files will become visible two weeks before the submission deadline.

## [Empty contract templates with the correct function interface are provided here.](https://github.com/Zvezdin/SolidityCourse/blob/master/10. Final-Project-Live-Defense/interface.sol)

I recommend to **keep the same function names, parameters and return value types**. **Anything else about the contract can be changed**, including:

* Function modifiers
* Adding other internal/external functions

**If you have a better solution** in mind that requires a different interface or are unable to find a solution fitting this interface, **feel free to change anything you need**, including function names and arguments. The common interface is only so that I can run unit tests on your code, testing the functionality and edge cases.