Smart Contract Implementation Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Days** | **Implementation** | **Start** | **Finish** |
| 1 | *1 Week* | **Make frontend structure in Ethereum** | 1-1-2020 | 1-7-2020 |
| 2 | *2 weeks* | **Learning and Making wallet || each user Account in JavaScript** | 1-8-2020 | 1-14-2020 |
| 3 | *3 weeks* | **Learning Design Language web3 JS** | 1-15-2020 | 2-6-2020 |
| 4 | *2 Weeks* | **Making Design in Web3 JS** | 2-7-2020 | 2-21-2020 |
|  | 1week | Making web3 JS Structure and View Design | 2-7-2020 | 2-14-2020 |
|  | 1week | Making Transaction System | 2-15-2020 | 2-21-2020 |
| 5 | *3 Weeks* | **Learn solidity language for contract** | 2-22-2020 | 3-13-2020 |
| 6 | *2 Weeks* | **Making a property or insurance contract in solidity** | 3-14-2020 | 3-28-2020 |
| 7 | *1 Weeks* | **Assemble frontend contract and wallet in Ethereum** | 3-29-2020 | 4-6-2020 |
| 8 | *1 Week* | **Deploy the contract on Ethereum Wallet** | 4-7-2020 | 4-14-2020 |
| 8 | *1 Week* | **Transaction to Execute Smart Contract** | 4-15-2020 | 4-21-2020 |

Complex

* [Poorly-coded smart contract](https://www.coindesk.com/understanding-dao-hack-journalists/) came from the Decentralized Autonomous Organization (DAO). A more recent bug also was discovered in the [smart contract used by Parity](https://cryptovest.com/news/parity-was-warned-in-august-about-bug-that-froze-over-500-wallets/), which was exploited and resulted in the loss of half a million Ether (ETH), worth more than $169 million.
* Security, specifically integer overflow, is also problematic when dealing with smart contracts written in Solidity.
* Public/private key algorithms are typically very slow and prohibitively expensive on the Ethereum network. Many classes of applications, such as most “data marketplaces”, are practically impossible on Ethereum. With libENI, however, those operations take milliseconds (10,000 times faster) and require very low gas fees (a few cents, compare to the aforementioned $500 in gas fees to perform a [cross-chain transaction](https://en.bitcoin.it/wiki/Atomic_cross-chain_trading)).

Innovative

* E-government, banking, insurance, energy, telecommunication, but also in the music & film industry, the art world, mobility, education
* For governments, smart contracts could provide an infinitely more secure system to vote online, that way hiking low voter turnouts.
* Banks use smart contracts to log change of ownership and transfer payments.

Applicable

* **Supply Chain** – Logistics companies can use smart contracts to update their suppliers and inventories based on deliveries.
* **Real Estate**  Real estate transactions way less stressful. The buyer and the seller draw a smart contract that’ll transfer the ownership of the property to the buyer on receiving the agreed upon amount.
* **Insurance** Health insurance can be disbursed via a smart contract on receiving the confirmation (Proof of Work) of a surgery or health procedure.

Significant Scope

* **Smart contract** is to provide security while transaction and reduce surplus transaction costs. It saves time and conflict and is also cheaper, faster and more secure way of payment as compared to the traditional system.
* It will allow the performance of credible transactions without third parties.
* Which ensures the safest transactions using multiple nodes.

Suitable size

Solidity contracts are compiled to EVM bytecode. Contract bytecode costs 200 gas per byte of bytecode, and since block gas limit is 8 million right now, you could theoretically get a ~40,000 (a little less since there's extra for deployment/tx data) byte contract deployed. It's unlikely that you'd be able to create a contract this size though since blocks don't usually have that much extra space already. Regarding execution speed, this is totally dependent on your implementation.

There's actually a 32kb limit per transaction as explained here. That page also says there's also a 24576 byte limit for contract size, so I stand corrected! The EIP for the contract byte limit can be found here.

In Solidity, the maximum size of a contract is restricted to 24 KB by [EIP 170](https://github.com/ethereum/EIPs/blob/master/EIPS/eip-170.md). If the same code is inlined multiple times, it adds up in size and that size limit can be hit easily.