

Patient Tasks and Storage:

- The patient will register and add his public key which will be stored in SC(smart contract).
 - ``mapping(address => string) publicKey``
- The patient will add the data (text field in the form) which he wants to store. The data will be hashed to create a unique identifier for repeated data and the data will be encrypted using the patient's publicKey to further decrypt it.
 - ``Struct DATA {string hash; string data; string prescription}``
 - ``mapping(address => string) patientDataHash``
- The patient can send the data to any doctor using the doctor's address. The public key of the doctor will be fetched from the smart contract.
 - The patient will add the hash of the data he wants to send, his private key (to decrypt the data), and the doctor's address.
 - The data will be decrypted and then will again encrypt using the doctor's public key so that doctor can decrypt the data and hence developing a secure transfer fo data.

Doctor Tasks and Storage:

- The doctor will register and add his public key which will be stored in SC(smart contract).
- The doctor will add the prescription (text field in the form) to send to the patient and also the unique hash identifier of the data. The prescription will be encrypted using the patient's and doctor's public key to make a copy for both of them and the prescription will be sent to the patient.

