Three roles:

admin

user

provider

Admin:

Admin deploy the contract

Move the ownership

Can withdraw the fee

Only admin can kill the contract

User:

Can register to a service

Can unsubscribe from a service

Can pay bill

Can move the funds to another user

Provider:

Can register as a provider

Can unsubscribe as a provider

Can see all the users under him

Can set the bill amount to a user

Admin:

Contract name: ownersContract

Variables:

Address owner;

function ownersContract() – Set the ownership to admin

modifier onlyOwner() – Checks if the function is executed by owner

function kill() – kills the contract

function transferOwnership() – Transfers the ownership

function withdrawFee() – Withdraw the fee from the contract and move it to the owners address

**Provider:**

Contract name: providersContract() is ownersContract

**Variables:**

address provider;

enum providerState {inactive, active, blocked} – Stores the provider’s state

Struct providerDetails {

String serviceName,

String companyName,

String cntactLastName,

String contactFirstName,

String contactPhone,

Bool active

}

mapping (provider => modDetails) as providerInfo; - Stores the provider’s details.

mapping(uint => address) public usersInfo; - Stores all the users who are registered with this provider.

**Functions:**

function registerProvider(provider) - Register as a provider

function unregisterProvider(provider) - Unregister as a provider

function getUsers() - Can see all the users registered for the service

function setBill(address user) - Can set the bill amount to a user

**User:**

**Variables:**

enum state {inactive, active, blocked} – Stores the state of the user

Struct userDetails {

String userLastName,

String userFirstName,

String userPhone,

State userState;

Uint providerCount; - Stores the count of the providers to which the user is registered to

} – Stores the details of the user

Mapping (address -> userDetails) as userInfo – Retrieve the user details using user address

struct allProviders {

address[] registeredTo;

} – Stores the addresses of the providers who registered

mapping (address => allProviders) userToProviders; - Mapping to retrieve all provider addresses using the user address.

**Functions:**

function addUser(string \_userLastName, string \_userFirstName, string \_userPhone) public returns (string)

* adds user to the system
* check if the user already exits

function removeUser(address \_user)

* removes the user (makes the state to inactive)

function regToService(address provider) - Can register to a service

* Check if the service provider is available and active

function unregToService(address provider) - Can unsubscribe from a service

function payBill(uint256 amount) - Can pay bill

function transferFund(address anotheruser, uint256 amount) - Can move the funds to another user