**Application overview:**

This application is designed to demonstrate how Medical history of a person can be modeled on the Blockchain. In the scenario members are modeled using Blockchain technology with the following attributes:

| **Attribute** | **Type** |
| --- | --- |
| ILNSID | Unique string formed of two chars followed by a 7 digit int, used as the key to identify the Illness a person had and its cause/report details. |
| Name | Identity of participant |
| DOB | String |
| Gender | String |
| Weight | 2 digit int |
| Blood Grp | String |
| Dead | Boolean |
| Status | Int between 0 and 4 |

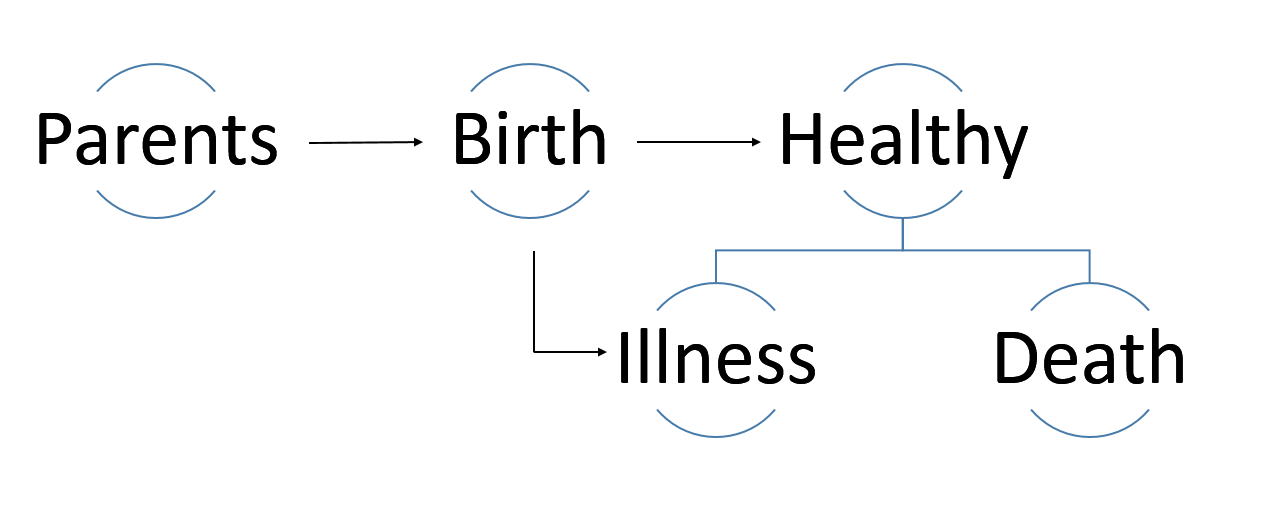
The application is designed to allow participants to interact with the member medical history creating, updating and transferring them as their permissions allow. The stages included in the application are as follows:

| **Stage** | **Permissions** |
| --- | --- |
| Parents | Create, Read ,Transfer |
| Birthday | Read ,Update ,Transfer |
| Healthy | Read ,Update ,Transfer |
| Illness | Read ,Update ,Transfer |
| Death | Read ,Update |

The demonstration allows a view of the ledger that stores all the interactions that the above stages have has with the respective member. The ledger view shows the parent every transaction that has occurred showing who tried to do what at what time and to which member. The ledger view also allows the user to see transactions that they were involved with as well as showing the interactions with the members e.g. they can see when it was created.

**Application scenario:**

The scenario goes through the lifecycle of a person which has the following stages:



**Stages:**

1. A person when born will have the Medical history of the **Parents** (including the sicknesses while the child was being carried).
2. When the child was given **Birth**, the Medical history of the baby when born was taken, the health status (any allergies, complications…)
3. The **Healthy** stage of a member is when he/she is at full health state.
4. When the member gets sick due to any reason, the details of the Illness will be noted in the **Illness** stage
5. There will be multiple entries in the Illness stage respectively whenever the person gets sick. Each Illness will be identified with a unique ID - ILNSID.
6. The final stage is **Death**, this stage can either be reached due to prolonged sickness or due to calamities from a healthy stage.
7. The final Death stage will have the details of the cause of the death and the treatment given to avoid it.

**Scenarios:**

Multiple scenarios shown below were covered while designing:

**Parents to Birth:**

This is the case when the person was born and the transfer happens from Parents to Birth containing all the medical history details of parents, the medical conditions during the carrying of the child and until the birth.

**Birthday to Healthy:**

This is the case when the new born baby gets to the first healthiest point and all the details like the born weight, BP, pulse rate will be transferred.

**Healthy to Illness:**

This is when the person gets sick and the cause for it and any earlier similar situations can be obtained.

**Illness to Heathy:**

During this scenario, the details like the treatment given to cure the Illness can be entered.

**Illness to Death:**

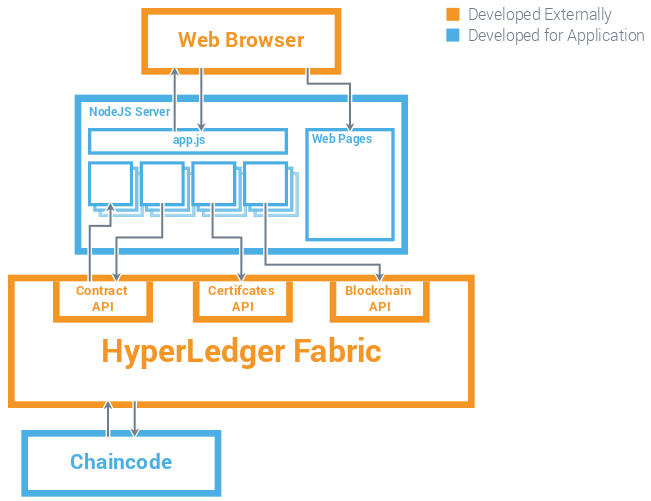
This is when Illness prolongs and leads to the death and all the details about the Illness, treatment can be entered.

**Healthy to Death:**

In this scenario, the details like the cause of the death if it is sudden, i.e., the person goes from a full health to the death.

**Component model:**

The demo will be built using a 3 tier architecture. The user interacts with the demo using a [web front end](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Documentation/Client%20Side.md) that is provided by the NodeJS server in the middle tier. This web front end uses JavaScript to make HTTP requests to the NodeJS server which has an API ([defined here](Documentation/API Methods.md)) which in turn makes calls via HTTP to the HyperLedger fabric to get details about the blockchain and also interact with the [chaincode](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Chaincode/src/vehicle_code/vehicles.go).

[](https://github.com/IBM-Blockchain/car-lease-demo/blob/master/Images/Technical_Component_Model.png)

Note: The demo currently does not include the ability to create, transfer or update the details of the Medical history of a member, but shows the entire flow from since a member was born until the death.

The code is developed in go language and is written to handle the above mentioned scenarios.

The URL to the GIThub repository:

<https://github.com/ravivarmakv/Medical_History_Recorder>

The URL to the GITLab repository:

<https://git.ng.bluemix.net/IT2017_TEAM17041115303100584/it2017_team17041115303100584/tree/Medical_History_Recorder>