'use strict';

const { Contract } = require('fabric-contract-api');

class HealthcareContract extends Contract {

async initLedger(ctx) {

// Initialize the ledger with a sample patient record

const patientRecord = {

ID: 'patient1',

Name: 'John Doe',

Diagnosis: null,

Doctor: null,

State: 'registered'

};

await ctx.stub.putState(patientRecord.ID, Buffer.from(JSON.stringify(patientRecord)));

console.info(`INFO: Patient record ${patientRecord.ID} initialized`);

}

async getPatientRecord(ctx, patientID) {

const patientRecord = await ctx.stub.getState(patientID);

if (!patientRecord || patientRecord.length === 0) {

throw new Error(`The patient record ${patientID} does not exist`);

}

console.info(`Patient Record key: ${patientRecord.ID}`);

console.info(`Patient Record value: ${patientRecord}`);

ctx.stub.setEvent('getPatientRecord', patientRecord);

return JSON.stringify(patientRecord.toString());

}

// Transaction submitter: Hospital

async registerPatient(ctx, id, name) {

const submitter = ctx.stub.getCreator().mspid;

if (!submitter.includes('Hospital')) {

throw new Error(`Only the hospital can register a patient`);

}

const newPatientRecord = {

ID: id,

Name: name,

Diagnosis: null,

Doctor: null,

State: 'registered'

};

await ctx.stub.putState(id, Buffer.from(JSON.stringify(newPatientRecord)));

console.info(`New patient record ${newPatientRecord.ID} registered`);

return JSON.stringify(newPatientRecord);

}

// Transaction submitter: Doctor

async updateDiagnosis(ctx, patientID, newDiagnosis) {

const submitter = ctx.stub.getCreator().mspid;

if (!submitter.includes('Doctor')) {

throw new Error(`Only the doctor can update a patient's diagnosis`);

}

const patientRecordBuffer = await ctx.stub.getState(patientID);

const patientRecordString = patientRecordBuffer.toString();

const patientRecord = JSON.parse(patientRecordString);

patientRecord.Diagnosis = newDiagnosis;

patientRecord.Doctor = submitter;

await ctx.stub.putState(patientID, Buffer.from(JSON.stringify(patientRecord)));

}

// Transaction submitter: Hospital

async dischargePatient(ctx, patientID, newState) {

const submitter = ctx.stub.getCreator().mspid;

if (!submitter.includes('Hospital')) {

throw new Error(`Only the hospital can discharge a patient`);

}

const patientRecordBuffer = await ctx.stub.getState(patientID);

const patientRecordString = patientRecordBuffer.toString();

const patientRecord = JSON.parse(patientRecordString);

patientRecord.State = newState;

await ctx.stub.putState(patientID, Buffer.from(JSON.stringify(patientRecord)));

return patientRecord;

}

async deleteLedger(ctx, id) {

const patientRecord = await ctx.stub.getState(id);

if (!patientRecord || patientRecord.length === 0) {

throw new Error(`The patient record ${id} does not exist`);

}

return ctx.stub.deleteState(id);

}

}

module.exports = HealthcareContract;