pragma solidity ^0.4.25;

contract CreditCert {

//event

event stuInfoInitEvent (

uint64 stuID,

string stuName,

string usName,

string major,

string extInfo,

uint64 registerTime

);

event stuInfoUpdateEvent (

uint64 stuID,

string stuName,

string usName,

string major,

string extInfo,

uint64 updateTime

);

event stuGradeRecordEvent (

uint64 stuID,

string stuName,

uint32 grades,

uint32 averageGrade,

string extInfo,

uint64 recordTime

);

event stuGradeUpdateEvent (

uint64 stuID,

string stuName,

uint32 grades,

uint32 averageGrade,

string extInfo,

uint64 updateTime

);

event activityRegisterEvent (

uint32 actID,

string actName,

string sponsor,

string status,

string extInfo,

uint64 registerTime

);

event activityInfoUpdateEvent (

uint32 actID,

string actName,

string sponsor,

string status,

string extInfo,

uint64 updateTime

);

event activityGradeRecodeEvent (

uint32 actID,

uint64 stuID,

string actName,

string stuName,

string extInfo,

uint64 recordTime,

string actSignature

);

event certInfoInitEvent (

uint64 certID,

uint64 stuID,

string stuName,

string usName,

string studyTime,

uint64 initTime

);

event certInfoUpdateEvent (

uint64 certID,

uint64 stuID,

string stuName,

string usName,

string studyTime,

string certStatus,

uint64 updateTime,

string certSignature

);

//学生初始信息

struct stuInfo{

uint64 stuID; //学生唯一ID

string stuName; //学生姓名

string usName; //所在学校名字

uint32 usLevel; //学校星级 1-3

string major; //学生专业

string extInfo; //附加信息

uint64 time; //操作时间

uint32 grades; //学生年级

mapping (uint32 => gradeInfo) gradeInfoList; //学期成绩列表

mapping (uint32 => activityRecord) activityRecordList; //参与活动列表

}

//学期成绩信息，,增加必修学分和选修学分两项，用于展示证书信息审核功能

struct gradeInfo{

uint64 stuID; //学生唯一ID

string stuName; //学生姓名

uint32 grades; //学年，一般为1-4

uint32 averageGrades; //加权平均成绩

uint32 obligatoryCredit; //必修学分，新增

uint32 optionalCredit; //选修学分，新增

string extInfo; //附加信息

uint64 time; //操作时间

}

//活动发布信息

struct activityInfo{

uint32 actID; //活动ID

string actName; //活动名称

string organizer; //举办方

string status; //活动状态，一般分为未开始、进行中、已结束3种

string extInfo; //附加信息

uint64 time; //操作时间

}

//学生活动记录数据结构

struct activityRecord{

uint32 actID; //活动ID

uint64 stuID; //学生ID

string actName; //活动名称

string stuName; //学生姓名

string extInfo; //附加信息

uint64 time; //操作时间

string actSignature; //参与活动信息签名

}

//学历证书数据结构

struct certInfo{

uint64 certID; //证书ID

uint64 stuID; //学生ID

string stuName; //学生姓名

string usName; //证书颁发学校名称

string major; //学生专业

string studyTime; //在校学习时间

string certStatus; //证书状态

string extInfo; //附加信息

uint64 time; //操作时间

string certSignature; //证书发布签名

}

//学生信息列表：ID到学生信息表的映射

mapping (uint64 => stuInfo) stuInfoList;

//学历证书信息列表：ID到学历证书信息的映射

mapping (uint64 => certInfo) certInfoList;

//活动信息列表：ID到结构体的映射

mapping (uint32 => activityInfo) activityInfoList;

//学生注册信息初始化

function stuInfoInit (

uint64 stuID,

string stuName,

string usName,

uint32 usLevel,

string major,

string extInfo,

uint64 registerTime,

uint32 grades

)

public

returns (uint32)

{

stuInfoList[stuID] = stuInfo(stuID, stuName, usName, usLevel, major, extInfo, registerTime, grades);

emit stuInfoInitEvent(stuID, stuName, usName, major, extInfo, registerTime);

return 1;

}

//学生信息更新

function stuInfoUpdate (

uint64 stuID,

string stuName,

string usName,

uint32 usLevel,

string major,

string extInfo,

uint64 updateTime,

uint32 grades

)

public

returns (uint32)

{

assert (stuInfoList[stuID].stuID != 0);

stuInfoList[stuID] = stuInfo(stuID, stuName, usName, usLevel, major, extInfo, updateTime, grades);

emit stuInfoUpdateEvent(stuID, stuName, usName, major, extInfo, updateTime);

return 1;

}

//学生成绩记录,增加必修学分和选修学分两项

function stuGradeRecord (

uint64 stuID,

string stuName,

uint32 grades,

uint32 averageGrade,

uint32 obligatoryCredit, //新增

uint32 optionalCredit, //新增

string extInfo,

uint64 recordTime

)

public

returns (uint32)

{

stuInfoList[stuID].gradeInfoList[grades] = gradeInfo(stuID, stuName, grades, averageGrade, obligatoryCredit, optionalCredit, extInfo, recordTime);

emit stuGradeRecordEvent(stuID, stuName, grades, averageGrade, extInfo, recordTime);

return 1;

}

// 学生成绩更新

function stuGradeUpdate (

uint64 stuID,

string stuName,

uint32 grades,

uint32 averageGrade,

uint32 obligatoryCredit, //新增

uint32 optionalCredit, //新增

string extInfo,

uint64 updateTime

)

public

returns (uint32)

{

assert (stuInfoList[stuID].gradeInfoList[grades].stuID != 0);

stuInfoList[stuID].gradeInfoList[grades] = gradeInfo(stuID, stuName, grades, averageGrade, obligatoryCredit, optionalCredit, extInfo, updateTime);

emit stuGradeUpdateEvent(stuID, stuName, grades, averageGrade, extInfo, updateTime);

return 1;

}

// 活动发布

function activityRegister (

uint32 actID,

string actName,

string organizer,

string status,

string extInfo,

uint64 registerTime

)

public

returns (uint32)

{

activityInfoList[actID] = activityInfo(actID, actName, organizer, status, extInfo, registerTime);

emit activityRegisterEvent(actID, actName, organizer, status, extInfo, registerTime);

return 1;

}

// 活动信息更新

function activityInfoUpdate (

uint32 actID,

string actName,

string organizer, //此处我把sponsor改为了organizer，之前各处也都修改了

string status,

string extInfo,

uint64 updateTime

)

public

returns (uint32)

{

assert (activityInfoList[actID].actID != 0);

activityInfoList[actID] = activityInfo(actID, actName, organizer, status, extInfo, updateTime);

emit activityInfoUpdateEvent(actID, actName, organizer, status, extInfo, updateTime);

return 1;

}

// 活动成绩记录

function activityGradeRecode (

uint32 actID,

uint64 stuID,

string actName,

string stuName,

string extInfo,

uint64 recordTime,

string actSignature

)

public

returns (uint32)

{

assert (stuInfoList[stuID].stuID != 0);

stuInfoList[stuID].activityRecordList[actID] = activityRecord(actID, stuID, actName, stuName, extInfo, recordTime, actSignature);

emit activityGradeRecodeEvent(actID, stuID, actName, stuName, extInfo, recordTime, actSignature);

return 1;

}

//学生获得学分查询,新增

function allCreditQuery(uint64 stuID)

public

constant

returns(uint32, uint32, uint32)

{

assert (stuInfoList[stuID].stuID != 0);

uint32 tempOblgCredits;

uint32 tempOptCredits;

uint32 i = 1;

while(stuInfoList[stuID].gradeInfoList[i].stuID != 0)

{

tempOblgCredits += stuInfoList[stuID].gradeInfoList[i].obligatoryCredit;

tempOptCredits += stuInfoList[stuID].gradeInfoList[i].optionalCredit;

i++;

}

uint32 tempSum = tempOblgCredits + tempOptCredits;

return (tempSum, tempOblgCredits, tempOptCredits);

}

//学分信息审核，新增

function creditInfoCheck (uint64 stuID)

public

constant

returns(uint32)

{

assert (stuInfoList[stuID].stuID != 0);

uint32 allOblgCredits;

uint32 allOptCredits;

uint32 sum;

(sum, allOblgCredits, allOptCredits) = allCreditQuery(stuID);

if(sum < 150 || allOblgCredits < 90 || allOptCredits < 60)

{

return 0;

}

return 1;

}

// 证书信息发布

function certInfoInit (

uint64 certID,

uint64 stuID,

string stuName,

string usName,

string major,

string studyTime,

string certStatus,

string extInfo,

uint64 initTime,

string certSignature

)

public

returns (uint32)

{

certInfoList[certID] = certInfo(certID, stuID, stuName, usName, major, studyTime, certStatus, extInfo, initTime, certSignature);

emit certInfoInitEvent(certID, stuID, stuName, usName, studyTime,initTime);

return 1;

}

//证书信息更新

function certInfoUpdate (

uint64 certID,

uint64 stuID,

string stuName,

string usName,

string major,

string studyTime,

string certStatus,

string extInfo,

uint64 updateTime,

string certSignature

)

public

returns (uint32)

{

assert (certInfoList[certID].certID != 0);

certInfoList[certID] = certInfo(certID, stuID, stuName, usName, major, studyTime, certStatus, extInfo, updateTime, certSignature);

emit certInfoUpdateEvent(certID, stuID, stuName, usName, studyTime, certStatus, updateTime, certSignature);

return 1;

}

// 学生信息查询

function stuInfoQuery (uint64 stuID)

public

constant

returns (uint64, string, string, uint32, string, string, uint64, uint32)

{

assert (stuInfoList[stuID].stuID != 0);

stuInfo tempStuInfo = stuInfoList[stuID];

return (

tempStuInfo.stuID,

tempStuInfo.stuName,

tempStuInfo.usName,

tempStuInfo.usLevel,

tempStuInfo.major,

tempStuInfo.extInfo,

tempStuInfo.time,

tempStuInfo.grades);

}

// 学生成绩查询

function stuGradeQuery (uint64 stuID, uint32 grades)

public

constant

returns (uint64, string, uint32, uint32, string, uint64)

{

assert (stuInfoList[stuID].gradeInfoList[grades].stuID != 0);

gradeInfo tempGradeInfo = stuInfoList[stuID].gradeInfoList[grades];

return (

tempGradeInfo.stuID,

tempGradeInfo.stuName,

tempGradeInfo.grades,

tempGradeInfo.averageGrades,

tempGradeInfo.extInfo,

tempGradeInfo.time);

}

// 学生活动查询

function stuActQuery (uint64 stuID, uint32 actID)

public

constant

returns (uint32, uint64, string, string, string, uint64, string)

{

assert (stuInfoList[stuID].activityRecordList[actID].stuID != 0);

activityRecord tempActRecord = stuInfoList[stuID].activityRecordList[actID];

return (

tempActRecord.actID,

tempActRecord.stuID,

tempActRecord.actName,

tempActRecord.stuName,

tempActRecord.extInfo,

tempActRecord.time,

tempActRecord.actSignature);

}

// 学生证书查询

function stuCertQuery (uint64 certID)

public

constant

returns (uint64, uint64, string, string, string, string, string, string, uint64)

// returns (uint64, uint64, string, string, string, string, string, string, uint64, string)

{

assert (certInfoList[certID].certID != 0);

certInfo tempCertInfo = certInfoList[certID];

return (

tempCertInfo.certID,

tempCertInfo.stuID,

tempCertInfo.stuName,

tempCertInfo.usName,

tempCertInfo.major,

tempCertInfo.studyTime,

tempCertInfo.certStatus,

tempCertInfo.extInfo,

tempCertInfo.time

// tempCertInfo.certSignature

);

}

// 活动信息查询

function actInfoQuery (uint32 actID)

public

constant

returns (uint32, string, string, string, string, uint64)

{

assert (activityInfoList[actID].actID != 0);

activityInfo tempActInfo = activityInfoList[actID];

return (

tempActInfo.actID,

tempActInfo.actName,

tempActInfo.organizer,

tempActInfo.status,

tempActInfo.extInfo,

tempActInfo.time);

}

// 学历证书签名查询,新增，用于证书认证

function certSignatureQuery(uint64 certID)

public

constant

returns(string, string)

{

assert (certInfoList[certID].certID != 0);

return (certInfoList[certID].certSignature, certInfoList[certID].extInfo);

}

//总平均成绩查询

function averageGradeQuery(uint64 stuID) public returns(uint32){

uint32 averageGrade = 0;

uint32 sumGrade = 0;

for(uint32 i = stuInfoList[stuID].grades; i > 0; i--){

sumGrade = sumGrade + stuInfoList[stuID].gradeInfoList[i].averageGrades;

}

averageGrade = sumGrade/stuInfoList[stuID].grades;

return averageGrade;

}

//信用评估

function creditEvaluation(uint64 stuID) public constant returns(uint32, uint32, uint32, uint32){

assert (stuInfoList[stuID].stuID != 0);

uint32 score = getM1(stuID) + getM2(stuID) + getM3(stuID);

return (

getM1(stuID), //学校等级得分

getM2(stuID), //平均成绩得分

getM3(stuID), //年级得分

score //总信用评分

);

}

//学校等级得分

function getM1(uint64 stuID) public returns(uint32){

uint32 usLevel = stuInfoList[stuID].usLevel;

if(usLevel == 3)

return 70;

else if(usLevel == 2)

return 80;

else if(usLevel == 1)

return 90;

else

return 60;

}

//平均成绩得分

function getM2(uint64 stuID) public returns(uint32){

uint32 averageGrade = averageGradeQuery(stuID);

return averageGrade;

}

//年级得分

function getM3(uint64 stuID) public returns(uint32){

uint32 grades= stuInfoList[stuID].grades;

uint32 M3 = grades \* 10 + 50;

return M3;

}

}